

[illegible][illegible]

```
CCCCCCCC 000000 NN NN SSSSSSSS 000000 LL EEEEEEEEE
CCCCCCCC 000000 NN NN SSSSSSSS 000000 LL EEEEEEEEE
CC 00 00 NN NN SS SSSSSSSS 00 00 LL EE
CC 00 00 NN NN SS SSSSSSSS 00 00 LL EE
CC 00 00 NNNN NN SS SSSSSSSS 00 00 LL EE
CC 00 00 NNNN NN SS SSSSSSSS 00 00 LL EE
CC 00 00 NN NN NN SSSSSSSS 00 00 LL EE
CC 00 00 NN NN NN SSSSSSSS 00 00 LL EE
CC 00 00 NN NN NN SSSSSSSS 00 00 LL EE
CC 00 00 NN NN NN SSSSSSSS 00 00 LL EE
CC 00 00 NN NN NN SSSSSSSS 00 00 LL EE
CCCCCCCC 000000 NN NN SSSSSSSS 000000 LLLLLLLLLL EEEEEEEEE
CCCCCCCC 000000 NN NN SSSSSSSS 000000 LLLLLLLLLL EEEEEEEEE
                                ....
                                ....
                                ....
                                ....

LL I I I I I SSSSSSSS
LL I I I I I SSSSSSSS
LL I I I I I SS
LL I I I I I SS
LL I I I I I SS
LL I I I I I SS
LL I I I I I SSSSSSS
LL I I I I I SSSSSSS
LL I I I I I SS
LL I I I I I SS
LL I I I I I SS
LL I I I I I SS
LLLLLLLLLLLL I I I I I SSSSSSSS
LLLLLLLLLLLL I I I I I SSSSSSSS
```

```
0001 0 MODULE console (IDENT = 'V04-000') =
0002 1 BEGIN
0003 1
0004 1
0005 1 *****
0006 1 *
0007 1 *   COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0008 1 *   DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0009 1 *   ALL RIGHTS RESERVED.
0010 1 *
0011 1 *   THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0012 1 *   ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0013 1 *   INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0014 1 *   COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0015 1 *   OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0016 1 *   TRANSFERRED.
0017 1 *
0018 1 *   THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0019 1 *   AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0020 1 *   CORPORATION.
0021 1 *
0022 1 *   DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0023 1 *   SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0024 1 *
0025 1 *
0026 1 *****
0027 1
0028 1 ++
0029 1 FACILITY: DECnet V2.0 event logger
0030 1
0031 1 ABSTRACT:
0032 1
0033 1   This module contains all console display formatting routines
0034 1   for the event receiver.
0035 1
0036 1 ENVIRONMENT:
0037 1
0038 1   VAX/VMS operating system. unprivileged user mode,
0039 1
0040 1 AUTHOR: Tim Halvorsen, June 1980
0041 1
0042 1 Modified by:
0043 1
0044 1   V03-019 MKP0001      Kathy Perko      12-June-1984
0045 1   Add Ethernet address to Network Management events. And
0046 1   increase size of OPCOM message buffers.
0047 1
0048 1   V03-018 PRB0327      Paul Beck      5-Apr-1984  22:14
0049 1   No longer suppress display of area 1.
0050 1
0051 1   V017   TMH0017      Tim Halvorsen      20-Sep-1983
0052 1   Add support for DECnet/PLUTO events (simply avoid
0053 1   printing "Parameter #n =" for these events).
0054 1   Fix printing area numbers for area #1.
0055 1
0056 1   V016   TMH0016      Tim Halvorsen      26-Apr-1983
0057 1   Add "verification password required from PH3 node" and
```


58	0058	1		"dropped by adjacent node" reasons.
59	0059	1		
60	0060	1	V015	TMH0015 Tim Halvorsen 08-Apr-1983
61	0061	1		Make ECOs approved during the March 1983 DRG meetings.
62	0062	1		Add support for DECnet/SNA Gateway events (simply avoid
63	0063	1		printing "Parameter #n =" for these events).
64	0064	1		Suppress printing area numbers for area #1.
65	0065	1		
66	0066	1	V014	TMH0014 Tim Halvorsen 28-Nov-1982
67	0067	1		Add extra blank line after each formatted event,
68	0068	1		to help make each one more readable.
69	0069	1		
70	0070	1	V013	TMH0013 Tim Halvorsen 16-Nov-1982
71	0071	1		Display area addresses correctly.
72	0072	1		Add formatting for new NODE and DTE event parameters
73	0073	1		for event class 0.
74	0074	1		
75	0075	1	V012	TMH0012 Tim Halvorsen 08-Nov-1982
76	0076	1		Fix display of DTE old/new state.
77	0077	1		
78	0078	1	V011	TMH0011 Tim Halvorsen 26-Sep-1982
79	0079	1		Add X.25 counters and VMS-specific events.
80	0080	1		Add "module" to the list of entities which
81	0081	1		are identified before the actual parameters.
82	0082	1		
83	0083	1	V010	TMH0010 Tim Halvorsen 29-Jul-1982
84	0084	1		Add Phase IV event parameters.
85	0085	1		
86	0086	1	V009	TMH0009 Tim Halvorsen 20-Jun-1982
87	0087	1		Update list of line and circuit counters.
88	0088	1		
89	0089	1	V008	TMH0008 Tim Halvorsen 28-Dec-1981
90	0090	1		Add PCL-11B and UNA counters.
91	0091	1		
92	0092	1	V007	TMH0007 Tim Halvorsen 19-Aug-1981
93	0093	1		Change text of CRC remote node parameter.
94	0094	1		
95	0095	1	V006	TMH0006 Tim Halvorsen 05-Aug-1981
96	0096	1		Add DAP CRC event formatting
97	0097	1		
98	0098	1	V005	TMH0005 Tim Halvorsen 08-Jul-1981
99	0099	1		Add PCL11-B line counters, and new 2.2 events and parameters.
100	0100	1		Increase maximum size of output strings sent to OPCOM (monitor)
101	0101	1		to 512, as a result of enhancements made to OPCOM and \$BRDCST.
102	0102	1		
103	0103	1	V004	TMH0004 Tim Halvorsen 07-Jul-1981
104	0104	1		Use SHRLIB\$ rather than LIB\$ for NMALIBRY.L32.
105	0105	1		
106	0106	1	V003	TMH0003 Tim Halvorsen 05-Jan-1981
107	0107	1		Fix decoding of line service status and display
108	0108	1		error message associated with status.
109	0109	1		
110	0110	1	V002	TMH0002 Tim Halvorsen 19-Nov-1980
111	0111	1		Make formatting routines callable with an action
112	0112	1		routine to actually perform the output. This allows
113	0113	1		the monitor sink output routine to use the formatting
114	0114	1		routines as well as the console sink.

```

: 115      0115 1 |
: 116      0116 1 |          V001      TMH0001      Tim Halvorsen  17-Nov-1980
: 117      0117 1 |          Remove noise words in formatted event output to
: 118      0118 1 |          reduce overall size of the output message so that
: 119      0119 1 |          it won't be truncated by OPCOM.
: 120      0120 1 |          Write the formatted output to SYS$OUTPUT if we are
: 121      0121 1 |          logging received events.
: 122      0122 1 |          --
: 123      0123 1 |
: 124      0124 1 |          Include files
: 125      0125 1 |
: 126      0126 1 |
: 127      0127 1 |
: 128      0128 1 | LIBRARY 'SYSS$LIBRARY:STARLET';          ! VAX/VMS common definitions
: 129      0129 1 |
: 130      0130 1 | LIBRARY 'SHRLIB$:NMALIBRY';              ! Get counter type definitions
: 131      0131 1 |
: 132      0132 1 | LIBRARY 'LIB$:EVCDEF';                  ! Event data definitions
: 133      0133 1 |
: 134      0134 1 | REQUIRE 'LIB$:EVLDEF';                  ! Facility definitions

```

```
136 0543 1 |
137 0544 1 | Table of contents
138 0545 1 |
139 0546 1 |
140 0547 1 | FORWARD ROUTINE
141 0548 1 |     format_event:      NOVALUE,      | Format an event record
142 0549 1 |     append_strings:    NOVALUE,      | Append a string to another string
143 0550 1 |     format_parameter:  NOVALUE,      | Format a NICE parameter
144 0551 1 |     unknown_parameter: NOVALUE,      | Format an unknown parameter
145 0552 1 |     unknown_param_data: NOVALUE,     | Format unknown parameter data
146 0553 1 |     format_counter:    NOVALUE,      | Format a NICE counter
147 0554 1 |     format_nodeadr:    NOVALUE,      | Format a area/node address
148 0555 1 |
149 0556 1 |
150 0557 1 | Define macro for message reporting
151 0558 1 |
152 0559 1 |
153 0560 1 | MACRO
154 M 0561 1 |     msg(ident) =
155 M 0562 1 |         BEGIN
156 M 0563 1 |             %IF NOT %DECLARED(%NAME('evl$',ident))
157 M 0564 1 |                 %THEN EXTERNAL LITERAL %NAME('evl$',ident); %FI
158 M 0565 1 |             %NAME('evl$',ident)
159 0566 1 |         END%;
160 0567 1 |
161 0568 1 |
162 0569 1 | Define a macro for constructing output strings
163 0570 1 |
164 0571 1 |
165 0572 1 | MACRO
166 M 0573 1 |     append(string) =
167 M 0574 1 |         append_strings(desc,UPLIT BYTE(%ASCII string)
168 0575 1 |             %IF %LENGTH GTR 1 %THEN ,%REMAINING %FI)%);
169 0576 1 |
170 0577 1 |
171 0578 1 | Literals
172 0579 1 |
173 0580 1 |
174 0581 1 | LITERAL
175 0582 1 |     true = 1,
176 0583 1 |     false = 0,
177 0584 1 |     max_output = 512;      | Maximum length of output string
178 0585 1 |
179 0586 1 |
180 0587 1 | External storage
181 0588 1 |
182 0589 1 |
183 0590 1 | EXTERNAL
184 0591 1 |     evl$gl_logmask:      $BBLOCK;    | Internal logging mask
185 0592 1 |
186 0593 1 |
187 0594 1 | External routines
188 0595 1 |
189 0596 1 |
190 0597 1 | EXTERNAL ROUTINE
191 0598 1 |     evl$unjulian,        | Convert julian half-day time
192 0599 1 |     wkq$add_work_item;   | Add work to work queue
```



```

193      0600 1
194      0601 1
195      0602 1
196      0603 1
197      0604 1
198      0605 1
199      0606 1
200      0607 1
201      M 0608 1
202      M 0609 1
203      0610 1
204      0611 1
205      P 0612 1
206      P 0613 1
207      P 0614 1
208      P 0615 1
209      P 0616 1
210      P 0617 1
211      P 0618 1
212      P 0619 1
213      P 0620 1
214      P 0621 1
215      P 0622 1
216      P 0623 1
217      P 0624 1
218      P 0625 1
219      P 0626 1
220      P 0627 1
221      P 0628 1
222      P 0629 1
223      P 0630 1
224      P 0631 1
225      P 0632 1
226      P 0633 1
227      P 0634 1
228      P 0635 1
229      P 0636 1
230      P 0637 1
231      P 0638 1
232      P 0639 1
233      P 0640 1
234      P 0641 1
235      P 0642 1
236      P 0643 1
237      P 0644 1
238      P 0645 1
239      P 0646 1
240      P 0647 1
241      P 0648 1
242      P 0649 1
243      P 0650 1
244      P 0651 1
245      P 0652 1
246      P 0653 1
247      P 0654 1
248      P 0655 1
249      P 0656 1

      ! Define all known events and the associated message text
      !
      MACRO
      evtdecl [name] = %NAME('evl$_',name)%,
      evtdef [name] = WORD(%NAME('evc$_',name)),LONG(msg(name))%,
      event_list [] =
      EXTERNAL LITERAL evtdecl(%REMAINING);
      BIND known_events = UPLIT(evtdef(%REMAINING),LONG(0))%;

      event_list(
      nma_los,
      nma_anc,
      nma_alc,
      nma_als,
      nma_lcz,
      nma_ncz,
      nma_psl,
      nma_abs,
      nma_ctr,
      nma_zer,

      scl_lns,
      scl_acr,

      nsl_ims,
      nsl_ifc,
      nsl_dbr,

      tpl_apl,
      tpl_upl,
      tpl_rpl,
      tpl_opl,
      tpl_pfm,
      tpl_pru,
      tpl_vfr,
      tpl_ldf,
      tpl_cds,
      tpl_cdo,
      tpl_lup,
      tpl_ilf,
      tpl_isf,
      tpl_iof,
      tpl_rch,
      tpl_aup,
      tpl_arj,
      tpl_ach,
      tpl_lds,
      tpl_ldo,

      dll_lsc,
      dll_rsc,
      dll_prs,
      dll_snd,
      dll_ret,

```

```

250 P 0657 1 dll_slc,
251 P 0658 1 dll_bhf,
252 P 0659 1 dll_sad,
253 P 0660 1 dll_stt,
254 P 0661 1 dll_lbs,
255 P 0662 1 dll_rst,
256 P 0663 1 dll_stc,
257 P 0664 1 dll_rme,
258 P 0665 1 dll_ifl,
259 P 0666 1 dll_sfl,
260 P 0667 1 dll_rfl,
261 P 0668 1 dll_cdc,
262 P 0669 1 dll_dtu,
263 P 0670 1 dll_dtd,
264 P 0671 1
265 P 0672 1 pll_dsr,
266 P 0673 1 pll_rin,
267 P 0674 1 pll_car,
268 P 0675 1 pll_mem,
269 P 0676 1 pll_com,
270 P 0677 1 pll_pfm,
271 P 0678 1
272 P 0679 1 vms_dbc,
273 P 0680 1 vms_dpc,
274 P 0681 1 vms_dp2,
275 P 0682 1 vms_pcr,
276 0683 1 vms_ptr);
277 0684 1
278 0685 1
279 0686 1 Define all known counters and the associated message text
280 0687 1
281 0688 1
282 0689 1 MACRO
283 0690 1 ctrdecl [name] = %NAME('evl$',name)%,
284 0691 1 ctrdef [name] = WORD(%NAME('nma$%',name)),LONG(msg(name))%,
285 M 0692 1 counter_list [ ] =
286 M 0693 1 EXTERNAL LITERAL ctrdecl(%REMAINING);
287 0694 1 BIND known_counters = UPLIT(ctrdef(%REMAINING),LONG(0))%;
288 0695 1
289 P 0696 1 counter_list(
290 P 0697 1 ctnod_zer,
291 P 0698 1 ctnod_brc,
292 P 0699 1 ctnod_bsn,
293 P 0700 1 ctnod_mrc,
294 P 0701 1 ctnod_msn,
295 P 0702 1 ctnod_crc,
296 P 0703 1 ctnod_csn,
297 P 0704 1 ctnod_rto,
298 P 0705 1 ctnod_rse,
299 P 0706 1 ctnod_mll,
300 P 0707 1 ctnod_apl,
301 P 0708 1 ctnod_nul,
302 P 0709 1 ctnod_nol,
303 P 0710 1 ctnod_opl,
304 P 0711 1 ctnod_pfe,
305 P 0712 1 ctnod_rul,
306 P 0713 1 ctnod_ver,

```


307	P 0714	1	
308	P 0715	1	ctcir_zer.
309	P 0716	1	ctcir_apr.
310	P 0717	1	ctcir_dps.
311	P 0718	1	ctcir_acl.
312	P 0719	1	ctcir_crl.
313	P 0720	1	ctcir_tpr.
314	P 0721	1	ctcir_tps.
315	P 0722	1	ctcir_tcl.
316	P 0723	1	ctcir_ldn.
317	P 0724	1	ctcir_ifl.
318	P 0725	1	ctcir_brc.
319	P 0726	1	ctcir_bsn.
320	P 0727	1	ctcir_mby.
321	P 0728	1	ctcir_dbr.
322	P 0729	1	ctcir_dbs.
323	P 0730	1	ctcir_dei.
324	P 0731	1	ctcir_deo.
325	P 0732	1	ctcir_rrt.
326	P 0733	1	ctcir_lrt.
327	P 0734	1	ctcir_rbe.
328	P 0735	1	ctcir_lbe.
329	P 0736	1	ctcir_sie.
330	P 0737	1	ctcir_slr.
331	P 0738	1	ctcir_ubu.
332	P 0739	1	ctcir_rpe.
333	P 0740	1	ctcir_lpe.
334	P 0741	1	ctcir_lir.
335	P 0742	1	ctcir_rir.
336	P 0743	1	ctcir_nir.
337	P 0744	1	ctcir_mne.
338	P 0745	1	ctcir_eri.
339	P 0746	1	ctcir_ero.
340	P 0747	1	ctcir_rto.
341	P 0748	1	ctcir_lto.
342	P 0749	1	ctcir_ber.
343	P 0750	1	ctcir_bel.
344	P 0751	1	
345	P 0752	1	ctlin_apr.
346	P 0753	1	ctlin_dps.
347	P 0754	1	ctlin_acl.
348	P 0755	1	ctlin_tpr.
349	P 0756	1	ctlin_tps.
350	P 0757	1	ctlin_tcl.
351	P 0758	1	ctlin_ldn.
352	P 0759	1	ctlin_ifl.
353	P 0760	1	ctlin_brc.
354	P 0761	1	ctlin_bsn.
355	P 0762	1	ctlin_mby.
356	P 0763	1	ctlin_dbr.
357	P 0764	1	ctlin_dbs.
358	P 0765	1	ctlin_mbl.
359	P 0766	1	ctlin_bid.
360	P 0767	1	ctlin_bsl.
361	P 0768	1	ctlin_bsm.
362	P 0769	1	ctlin_dei.
363	P 0770	1	ctlin_deo.

```

: 364 P 0771 1 ctlin_rrt,
: 365 P 0772 1 ctlin_lrt,
: 366 P 0773 1 ctlin_rbe,
: 367 P 0774 1 ctlin_lbe,
: 368 P 0775 1 ctlin_sie,
: 369 P 0776 1 ctlin_slr,
: 370 P 0777 1 ctlin_sfl,
: 371 P 0778 1 ctlin_cdc,
: 372 P 0779 1 ctlin_rfl,
: 373 P 0780 1 ctlin_ufr,
: 374 P 0781 1 ctlin_ovr,
: 375 P 0782 1 ctlin_sbu,
: 376 P 0783 1 ctlin_ubu,
: 377 P 0784 1 ctlin_rpe,
: 378 P 0785 1 ctlin_lpe,
: 379 P 0786 1 ctlin_mbs,
: 380 P 0787 1 ctlin_msn,
: 381 P 0788 1 ctlin_rme,
: 382 P 0789 1 ctlin_lce,
: 383 P 0790 1 ctlin_mse,
: 384 P 0791 1
: 385 P 0792 1 ctxp_zer,
: 386 P 0793 1 ctxp_brc,
: 387 P 0794 1 ctxp_bsn,
: 388 P 0795 1 ctxp_blr,
: 389 P 0796 1 ctxp_bls,
: 390 P 0797 1 ctxp_crc,
: 391 P 0798 1 ctxp_csn,
: 392 P 0799 1 ctxp_fsr,
: 393 P 0800 1 ctxp_fss,
: 394 P 0801 1 ctxp_msa,
: 395 P 0802 1 ctxp_mca,
: 396 P 0803 1 ctxp_rse,
: 397 P 0804 1 ctxp_lir,
: 398 P 0805 1 ctxp_rir,
: 399 P 0806 1 ctxp_nir,
: 400 P 0807 1 ctxp_rst,
: 401 P 0808 1
: 402 P 0809 1 ctxs_zer,
: 403 P 0810 1 ctxs_mca,
: 404 P 0811 1 ctxs_icr,
: 405 0812 1 ctxs_llr);

```

```

407 0813 1 GLOBAL ROUTINE format_event (event, event_length,
408 0814 1 output_routine, output_parm): NOVALUE =
409 0815 1
410 0816 1 ---
411 0817 1
412 0818 1 Output an event record to the console sink.
413 0819 1
414 0820 1 Inputs:
415 0821 1
416 0822 1 event = Address of event record
417 0823 1 event_length = Length of event record
418 0824 1 output_routine = Address of routine to call for each formatted line
419 0825 1 output_parm = Optional parameter to be sent to output routine
420 0826 1
421 0827 1 Outputs:
422 0828 1
423 0829 1 None
424 0830 1 ---
425 0831 1
426 0832 2 BEGIN
427 0833 2
428 0834 2 MACRO
429 0835 2 new_line =
430 0836 2 BEGIN
431 0837 2 (.output_routine)(.output_parm, desc, .short_form); ! Call output action routine
432 0838 2 desc [0] = 0; ! Reset descriptor length
433 0839 2 ENDX;
434 0840 2
435 0841 2 MAP
436 0842 2 event: REF $BBLOCK; ! Address of event record
437 0843 2
438 0844 2 LOCAL
439 0845 2 ptr: REF $BBLOCK, ! Address of event data parameter
440 0846 2 days, seconds, msecs, ! For conversion of julian time
441 0847 2 comma_needed: BYTE, ! Boolean; true if comma needed on line
442 0848 2 time: VECTOR [2], ! Converted 64-bit system time
443 0849 2 short_form, ! Length of short form of display
444 0850 2 desc: VECTOR [2], ! Descriptor of output buffer
445 0851 2 buffer: VECTOR [max_output, BYTE]; ! Output buffer (extra for counters)
446 0852 2
447 0853 2 desc [0] = 0; ! Setup descriptor of buffer
448 0854 2 desc [1] = buffer;
449 0855 2 short_form = -1; ! Indicate no short form yet
450 0856 2
451 0857 2 append('DECnet event !UL.!UL', .event [evt$v_class], .event [evt$v_type]);
452 0858 2
453 0859 2 INCRA ptr FROM known_events BY 6 ! Scan known event table
454 0860 2 DO
455 0861 2 BEGIN
456 0862 2 MAP ptr: REF $BBLOCK;
457 0863 2 IF .ptr EQL 0 ! If end of table,
458 0864 2 THEN
459 0865 2 EXITLOOP; ! then exit with failure
460 0866 2 IF .event [evt$v_code] EQLU .ptr [0,0,16,0] ! If known event,
461 0867 2 THEN
462 0868 2 BEGIN
463 0869 2 LOCAL

```



```

464      0870      4      textbuf: VECTOR [128,BYTE], ! Message text buffer
465      0871      4      text: VECTOR [2]; ! Message text descriptor
466      0872      4      text [0] = 128; ! Setup descriptor
467      0873      4      text [1] = textbuf;
468      0874      4      $GETMSG(MSGID = .ptr [2,0,32,0], ! Get message text
469      0875      4      BUFADR = text,
470      0876      4      MSGLEN = text,
471      0877      4      FLAGS = 1); ! without any prefixing
472      0878      4      append(' !AS', text);
473      0879      4      EXITLOOP;
474      0880      4      END;
475      0881      4      END;
476      0882      4
477      0883      4      new_line; ! Start a new line
478      0884      4
479      0885      4      append('From node !AS', format_nodeadr(.event [evt$w_srcadr]));
480      0886      4
481      0887      4      IF .event [evt$b_srcnamlen] NEQ 0 ! If source node name supplied
482      0888      4      THEN
483      0889      4      append(' (!AC)', event [evt$b_srcnamlen]);
484      0890      4
485      0891      4      days = .event [evt$w_julian]; ! Get julian values
486      0892      4      seconds = .event [evt$w_seconds];
487      0893      4      msec = .event [evt$w_msecs];
488      0894      4
489      0895      4      evl$unjulian(days,seconds,msec,time); ! Convert to system quadword time
490      0896      4
491      0897      4      append(' !XD', time);
492      0898      4
493      0899      4      new_line; ! Start third line of message
494      0900      4      comma_needed = false; ! Start of line, no comma needed yet
495      0901      4
496      0902      4      ptr = event [evt$t_srcnam] + .event [evt$b_srcnamlen]; ! Point to event entity
497      0903      4
498      0904      4      SELECTU CH$RCHAR_A(ptr) ! Select entity based on entity ID
499      0905      4      OF
500      0906      4      SET
501      0907      4      [evc$c_src_lin]: BEGIN
502      0908      4      append('Line !AC', .ptr);
503      0909      4      ptr = CH$RCHAR_A(ptr) + .ptr;
504      0910      4      comma_needed = true;
505      0911      4      END;
506      0912      4
507      0913      4      [evc$c_src_cir]: BEGIN
508      0914      4      append('Circuit !AC', .ptr);
509      0915      4      ptr = CH$RCHAR_A(ptr) + .ptr;
510      0916      4      comma_needed = true;
511      0917      4      END;
512      0918      4
513      0919      4      [evc$c_src_nod]: BEGIN
514      0920      4      append('Node !AS', format_nodeadr(.ptr [0,0,16,0]));
515      0921      4      ptr = .ptr+2;
516      0922      4      IF CH$RCHAR(.ptr) GTRU 0
517      0923      4      THEN
518      0924      4      append(' (!AC)', .ptr);
519      0925      4      ptr = CH$RCHAR_A(ptr) + .ptr;
520      0926      4      comma_needed = true;

```

```
END;
[levc$src_are]: BEGIN
  append('Area !UB', .ptr [0,0,8,0]);
  ptr = .ptr+1;
  comma_needed = true;
END;
[levc$src_mod]: BEGIN
  append('Module !AC', .ptr);
  ptr = CH$RCHAR_A(ptr) + .ptr;
  comma_needed = true;
END;
TES;
WHILE .ptr LSSU .event + .event_length ! Until end of event record
DO
  BEGIN
    LOCAL
      startlen; ! Starting string length
    IF .desc [0] GTR 45 ! If line already too long,
    THEN
      BEGIN
        new_line; ! then start another line
        comma_needed = false; ! Indicate no comma on start of line
      END;
    startlen = .desc [0]; ! Save starting string length
    IF .comma_needed ! If comma needed,
    THEN
      BEGIN
        append(', '); ! then add it
        comma_needed = false; ! and reset flag
      END;
    IF .ptr [0,15,1,0] ! If counter data (bit 15),
    THEN
      BEGIN
        IF .short_form LSS 0 ! If first counter,
        THEN
          short_form = .startlen ! set short form up to this point
          ! and trailing comma
        ELSE
          short_form = 0; ! else skip all counter output
        new_line; ! start a new line
        format_counter(desc, ptr, ! then format the counter; update ptr
          .event + .event_length - .ptr);
      END
    ELSE
      format_parameter(desc, ptr, ! else format the parameter; update ptr
        .event + .event_length - .ptr,
        .event [evt$src_class]);
    IF .startlen NEQ .desc [0] ! If anything was appended to string,
    THEN
```

```

: 578      0984      3      comma_needed = true;
: 579      0985      3      END;
: 580      0986      3
: 581      0987      3      If .desc [0] GTR 0
: 582      0988      3      THEN
: 583      0989      3          new_line;
: 584      0990      3
: 585      0991      3      new_line;
: 586      0992      3
: 587      0993      3      END;

```

```

! then insert a comma

! If something in buffer,

! Output the last line

! Output one last blank line

```

```

.TITLE  CONSOLE
.IDENT  \V04-000\

.PSECT  $PLIT$,NOWRT,NOEXE,2

```

```

      0000 00000 P.AAA: .WORD 0
00000000G 00002 .LONG EVLS_NMA_LOS
      0001 00006 .WORD 1
00000000G 00008 .LONG EVLS_NMA_ANC
      0002 0000C .WORD 2
00000000G 0000E .LONG EVLS_NMA_ALC
      0003 00012 .WORD 3
00000000G 00014 .LONG EVLS_NMA_ALS
      0004 00018 .WORD 4
00000000G 0001A .LONG EVLS_NMA_LCZ
      0005 0001E .WORD 5
00000000G 00020 .LONG EVLS_NMA_NCZ
      0006 00024 .WORD 6
00000000G 00026 .LONG EVLS_NMA_PSL
      0007 0002A .WORD 7
00000000G 0002C .LONG EVLS_NMA_ABS
      0008 00030 .WORD 8
00000000G 00032 .LONG EVLS_NMA_CTR
      0009 00036 .WORD 9
00000000G 00038 .LONG EVLS_NMA_ZER
      0080 0003C .WORD 128
00000000G 0003E .LONG EVLS_SCL_LNS
      0081 00042 .WORD 129
00000000G 00044 .LONG EVLS_SCL_ACR
      00C0 00048 .WORD 192
00000000G 0004A .LONG EVLS_NSL_IMS
      00C1 0004E .WORD 193
00000000G 00050 .LONG EVLS_NSL_IFC
      00C2 00054 .WORD 194
00000000G 00056 .LONG EVLS_NSL_DBR
      0100 0005A .WORD 256
00000000G 0005C .LONG EVLS_TPL_APL
      0101 00060 .WORD 257
00000000G 00062 .LONG EVLS_TPL_UPL
      0102 00066 .WORD 258
00000000G 00068 .LONG EVLS_TPL_RPL
      0103 0006C .WORD 259
00000000G 0006E .LONG EVLS_TPL_OPL
      0104 00072 .WORD 260
00000000G 00074 .LONG EVLS_TPL_PFM

```


0105	00078	.WORD	261
00000000G	0007A	.LONG	EVL\$_TPL_PRU
0106	0007E	.WORD	262
00000000G	00080	.LONG	EVL\$_TPL_VFR
0107	00084	.WORD	263
00000000G	00086	.LONG	EVL\$_TPL_LDF
0108	0008A	.WORD	264
00000000G	0008C	.LONG	EVL\$_TPL_CDS
0109	00090	.WORD	265
00000000G	00092	.LONG	EVL\$_TPL_CDO
010A	00096	.WORD	266
00000000G	00098	.LONG	EVL\$_TPL_LUP
010B	0009C	.WORD	267
00000000G	0009E	.LONG	EVL\$_TPL_ILF
010C	000A2	.WORD	268
00000000G	000A4	.LONG	EVL\$_TPL_ISF
010D	000A8	.WORD	269
00000000G	000AA	.LONG	EVL\$_TPL_IOF
010E	000AE	.WORD	270
00000000G	000B0	.LONG	EVL\$_TPL_RCH
010F	000B4	.WORD	271
00000000G	000B6	.LONG	EVL\$_TPL_AUP
0110	000BA	.WORD	272
00000000G	000BC	.LONG	EVL\$_TPL_ARJ
0111	000C0	.WORD	273
00000000G	000C2	.LONG	EVL\$_TPL_ACH
0112	000C6	.WORD	274
00000000G	000C8	.LONG	EVL\$_TPL_LDS
0113	000CC	.WORD	275
00000000G	000CE	.LONG	EVL\$_TPL_LDO
0140	000D2	.WORD	320
00000000G	000D4	.LONG	EVL\$_DLL_LSC
0141	000D8	.WORD	321
00000000G	000DA	.LONG	EVL\$_DLL_RSC
0142	000DE	.WORD	322
00000000G	000E0	.LONG	EVL\$_DLL_PRS
0143	000E4	.WORD	323
00000000G	000E6	.LONG	EVL\$_DLL_SND
0144	000EA	.WORD	324
00000000G	000EC	.LONG	EVL\$_DLL_RET
0145	000F0	.WORD	325
00000000G	000F2	.LONG	EVL\$_DLL_SLC
0146	000F6	.WORD	326
00000000G	000F8	.LONG	EVL\$_DLL_BHF
0147	000FC	.WORD	327
00000000G	000FE	.LONG	EVL\$_DLL_SAD
0148	00102	.WORD	328
00000000G	00104	.LONG	EVL\$_DLL_STT
0149	00108	.WORD	329
00000000G	0010A	.LONG	EVL\$_DLL_LBS
014A	0010E	.WORD	330
00000000G	00110	.LONG	EVL\$_DLL_RST
014B	00114	.WORD	331
00000000G	00116	.LONG	EVL\$_DLL_STC
014C	0011A	.WORD	332
00000000G	0011C	.LONG	EVL\$_DLL_RME
014D	00120	.WORD	333

00000000G	00122	.LONG	EVL\$_DLL_IFL
014E	00126	.WORD	334
00000000G	00128	.LONG	EVL\$_DLL_SFL
014F	0012C	.WORD	335
00000000G	0012E	.LONG	EVL\$_DLL_RFL
0150	00132	.WORD	336
00000000G	00134	.LONG	EVL\$_DLL_CDC
0151	00138	.WORD	337
00000000G	0013A	.LONG	EVL\$_DLL_DTU
0152	0013E	.WORD	338
00000000G	00140	.LONG	EVL\$_DLL_DTD
0180	00144	.WORD	384
00000000G	00146	.LONG	EVL\$_PLL_DSR
0181	0014A	.WORD	385
00000000G	0014C	.LONG	EVL\$_PLL_RIN
0182	00150	.WORD	386
00000000G	00152	.LONG	EVL\$_PLL_CAR
0183	00156	.WORD	387
00000000G	00158	.LONG	EVL\$_PLL_MEM
0184	0015C	.WORD	388
00000000G	0015E	.LONG	EVL\$_PLL_COM
0185	00162	.WORD	389
00000000G	00164	.LONG	EVL\$_PLL_PFM
2000	00168	.WORD	8192
00000000G	0016A	.LONG	EVL\$_VMS_DBC
2001	0016E	.WORD	8193
00000000G	00170	.LONG	EVL\$_VMS_DPC
2002	00174	.WORD	8194
00000000G	00176	.LONG	EVL\$_VMS_DP2
2003	0017A	.WORD	8195
00000000G	0017C	.LONG	EVL\$_VMS_PCR
2004	00180	.WORD	8196
00000000G	00182	.LONG	EVL\$_VMS_PTR
00000000	00186	.LONG	0
	0018A	.BLKB	2
0000	0018C	.WORD	0
		P.AAB:	
00000000G	0018E	.LONG	EVL\$_CTNOD_ZER
0258	00192	.WORD	600
00000000G	00194	.LONG	EVL\$_CTNOD_BRC
0259	00198	.WORD	601
00000000G	0019A	.LONG	EVL\$_CTNOD_BSN
0262	0019E	.WORD	610
00000000G	001A0	.LONG	EVL\$_CTNOD_MRC
0263	001A4	.WORD	611
00000000G	001A6	.LONG	EVL\$_CTNOD_MSN
026C	001AA	.WORD	620
00000000G	001AC	.LONG	EVL\$_CTNOD_CRC
026D	001B0	.WORD	621
00000000G	001B2	.LONG	EVL\$_CTNOD_CSN
0276	001B6	.WORD	630
00000000G	001B8	.LONG	EVL\$_CTNOD_RTO
0280	001BC	.WORD	640
00000000G	001BE	.LONG	EVL\$_CTNOD_RSE
02BC	001C2	.WORD	700
00000000G	001C4	.LONG	EVL\$_CTNOD_MLL
0384	001C8	.WORD	900
00000000G	001CA	.LONG	EVL\$_CTNOD_APL

0385	001CE	.WORD	901
00000000G	001D0	.LONG	EVL\$ _CTNOD _NUL
0386	001D4	.WORD	902
00000000G	001D6	.LONG	EVL\$ _CTNOD _NOL
0387	001DA	.WORD	903
00000000G	001DC	.LONG	EVL\$ _CTNOD _OPL
038E	001E0	.WORD	910
00000000G	001E2	.LONG	EVL\$ _CTNOD _PFE
0398	001E6	.WORD	920
00000000G	001E8	.LONG	EVL\$ _CTNOD _RUL
03A2	001EC	.WORD	930
00000000G	001EE	.LONG	EVL\$ _CTNOD _VER
0000	001F2	.WORD	0
00000000G	001F4	.LONG	EVL\$ _CTCIR _ZER
0320	001F8	.WORD	800
00000000G	001FA	.LONG	EVL\$ _CTCIR _APR
0321	001FE	.WORD	801
00000000G	00200	.LONG	EVL\$ _CTCIR _DPS
0322	00204	.WORD	802
00000000G	00206	.LONG	EVL\$ _CTCIR _ACL
0325	0020A	.WORD	805
00000000G	0020C	.LONG	EVL\$ _CTCIR _CRL
032A	00210	.WORD	810
00000000G	00212	.LONG	EVL\$ _CTCIR _TPR
032B	00216	.WORD	811
00000000G	00218	.LONG	EVL\$ _CTCIR _TPS
032C	0021C	.WORD	812
00000000G	0021E	.LONG	EVL\$ _CTCIR _TCL
0334	00222	.WORD	820
00000000G	00224	.LONG	EVL\$ _CTCIR _LDN
0335	00228	.WORD	821
00000000G	0022A	.LONG	EVL\$ _CTCIR _IFL
03E8	0022E	.WORD	1000
00000000G	00230	.LONG	EVL\$ _CTCIR _BRC
03E9	00234	.WORD	1001
00000000G	00236	.LONG	EVL\$ _CTCIR _BSN
03EA	0023A	.WORD	1002
00000000G	0023C	.LONG	EVL\$ _CTCIR _MBY
03F2	00240	.WORD	1010
00000000G	00242	.LONG	EVL\$ _CTCIR _DBR
03F3	00246	.WORD	1011
00000000G	00248	.LONG	EVL\$ _CTCIR _DBS
03FC	0024C	.WORD	1020
00000000G	0024E	.LONG	EVL\$ _CTCIR _DEI
03FD	00252	.WORD	1021
00000000G	00254	.LONG	EVL\$ _CTCIR _DEO
0406	00258	.WORD	1030
00000000G	0025A	.LONG	EVL\$ _CTCIR _RRT
0407	0025E	.WORD	1031
00000000G	00260	.LONG	EVL\$ _CTCIR _LRT
0410	00264	.WORD	1040
00000000G	00266	.LONG	EVL\$ _CTCIR _RBE
0411	0026A	.WORD	1041
00000000G	0026C	.LONG	EVL\$ _CTCIR _LBE
041A	00270	.WORD	1050
00000000G	00272	.LONG	EVL\$ _CTCIR _SIE
041B	00276	.WORD	1051

00000000G	00278	.LONG	EVL\$_CTCIR_SLT
0429	0027C	.WORD	1065
00000000G	0027E	.LONG	EVL\$_CTCIR_UBU
044C	00282	.WORD	1100
00000000G	00284	.LONG	EVL\$_CTCIR_RPE
044D	00288	.WORD	1101
00000000G	0028A	.LONG	EVL\$_CTCIR_LPE
04D8	0028E	.WORD	1240
00000000G	00290	.LONG	EVL\$_CTCIR_LIR
04D9	00294	.WORD	1241
00000000G	00296	.LONG	EVL\$_CTCIR_RIR
04DA	0029A	.WORD	1242
00000000G	0029C	.LONG	EVL\$_CTCIR_NIR
0A8D	002A0	.WORD	2701
00000000G	002A2	.LONG	EVL\$_CTCIR_MNE
0ABE	002A6	.WORD	2750
00000000G	002A8	.LONG	EVL\$_CTCIR_ERI
0ABF	002AC	.WORD	2751
00000000G	002AE	.LONG	EVL\$_CTCIR_ERO
0AC0	002B2	.WORD	2752
00000000G	002B4	.LONG	EVL\$_CTCIR_RTO
0AC1	002B8	.WORD	2753
00000000G	002BA	.LONG	EVL\$_CTCIR_LTO
0AC2	002BE	.WORD	2754
00000000G	002C0	.LONG	EVL\$_CTCIR_BER
0AC3	002C4	.WORD	2755
00000000G	002C6	.LONG	EVL\$_CTCIR_BEL
0320	002CA	.WORD	800
00000000G	002CC	.LONG	EVL\$_CTLIN_APR
0321	002D0	.WORD	801
00000000G	002D2	.LONG	EVL\$_CTLIN_DPS
0322	002D6	.WORD	802
00000000G	002D8	.LONG	EVL\$_CTLIN_ACL
032A	002DC	.WORD	810
00000000G	002DE	.LONG	EVL\$_CTLIN_TPR
032B	002E2	.WORD	811
00000000G	002E4	.LONG	EVL\$_CTLIN_TPS
032C	002E8	.WORD	812
00000000G	002EA	.LONG	EVL\$_CTLIN_TCL
0334	002EE	.WORD	820
00000000G	002F0	.LONG	EVL\$_CTLIN_LDN
0335	002F4	.WORD	821
00000000G	002F6	.LONG	EVL\$_CTLIN_IFL
03E8	002FA	.WORD	1000
00000000G	002FC	.LONG	EVL\$_CTLIN_BRC
03E9	00300	.WORD	1001
00000000G	00302	.LONG	EVL\$_CTLIN_BSN
03EA	00306	.WORD	1002
00000000G	00308	.LONG	EVL\$_CTLIN_MBY
03F2	0030C	.WORD	1010
00000000G	0030E	.LONG	EVL\$_CTLIN_DBR
03F3	00312	.WORD	1011
00000000G	00314	.LONG	EVL\$_CTLIN_DBS
03F4	00318	.WORD	1012
00000000G	0031A	.LONG	EVL\$_CTLIN_MBL
03F5	0031E	.WORD	1013
00000000G	00320	.LONG	EVL\$_CTLIN_BID

03F6	00324	.WORD	1014
00000000G	00326	.LONG	EVL\$CTLIN_BS1
03F7	0032A	.WORD	1015
00000000G	0032C	.LONG	EVL\$CTLIN_BSM
03FC	00330	.WORD	1020
00000000G	00332	.LONG	EVL\$CTLIN_DEI
03FD	00336	.WORD	1021
00000000G	00338	.LONG	EVL\$CTLIN_DEO
0406	0033C	.WORD	1030
00000000G	0033E	.LONG	EVL\$CTLIN_RRT
0407	00342	.WORD	1031
00000000G	00344	.LONG	EVL\$CTLIN_LRT
0410	00348	.WORD	1040
00000000G	0034A	.LONG	EVL\$CTLIN_RBE
0411	0034E	.WORD	1041
00000000G	00350	.LONG	EVL\$CTLIN_LBE
041A	00354	.WORD	1050
00000000G	00356	.LONG	EVL\$CTLIN_SIE
041B	0035A	.WORD	1051
00000000G	0035C	.LONG	EVL\$CTLIN_SLT
0424	00360	.WORD	1060
00000000G	00362	.LONG	EVL\$CTLIN_SFL
0425	00366	.WORD	1061
00000000G	00368	.LONG	EVL\$CTLIN_CDC
0426	0036C	.WORD	1062
00000000G	0036E	.LONG	EVL\$CTLIN_RFL
0427	00372	.WORD	1063
00000000G	00374	.LONG	EVL\$CTLIN_UFD
0428	00378	.WORD	1064
00000000G	0037A	.LONG	EVL\$CTLIN_OVR
0429	0037E	.WORD	1065
00000000G	00380	.LONG	EVL\$CTLIN_SBU
042A	00384	.WORD	1066
00000000G	00386	.LONG	EVL\$CTLIN_UBU
044C	0038A	.WORD	1100
00000000G	0038C	.LONG	EVL\$CTLIN_RPE
044D	00390	.WORD	1101
00000000G	00392	.LONG	EVL\$CTLIN_LPE
0A8D	00396	.WORD	2701
00000000G	00398	.LONG	EVL\$CTLIN_MBS
0A8E	0039C	.WORD	2702
00000000G	0039E	.LONG	EVL\$CTLIN_MSN
0ABE	003A2	.WORD	2750
00000000G	003A4	.LONG	EVL\$CTLIN_RME
0ABF	003A8	.WORD	2751
00000000G	003AA	.LONG	EVL\$CTLIN_LCE
0AC0	003AE	.WORD	2752
00000000G	003B0	.LONG	EVL\$CTLIN_MSE
0000	003B4	.WORD	0
00000000G	003B6	.LONG	EVL\$CTXP_ZER
03E8	003BA	.WORD	1000
00000000G	003BC	.LONG	EVL\$CTXP_BRC
03E9	003C0	.WORD	1001
00000000G	003C2	.LONG	EVL\$CTXP_BSN
03F2	003C6	.WORD	1010
00000000G	003C8	.LONG	EVL\$CTXP_BLR
03F3	003CC	.WORD	1011

21	20	74	6E	65	76	65	20	74	65	6E	43	45	44	14	00430	P.AAC:	.ASCII	<20>\DECnet event !UL.!UL\	
									4C	55	21	2E	4C	55	0043F				
									53	41	21	20	2C	05	00445	P.AAD:	.ASCII	<5>\, !AS\	
53	41	21	20	65	64	6F	6E	29	43	41	21	28	20	06	0044B	P.AAE:	.ASCII	<13>\From node !AS\	
									44	25	21	20	2C	05	00459	P.AAF:	.ASCII	<6>\ (!AC)\	
									43	41	21	20	2C	05	00460	P.AAG:	.ASCII	<5>\, !XD\	
									43	41	21	20	2C	05	00466	P.AAH:	.ASCII	<8>\Line !AC\	
									53	41	21	20	2C	05	0046F	P.AAI:	.ASCII	<11>\Circuit !AC\	
									43	41	21	20	2C	05	0047B	P.AAJ:	.ASCII	<8>\Node !AS\	
									43	41	21	20	2C	05	00484	P.AAK:	.ASCII	<6>\ (!AC)\	
									43	41	21	20	2C	05	0048B	P.AAL:	.ASCII	<8>\Area !UB\	
									43	41	21	20	2C	05	00494	P.AAM:	.ASCII	<10>\Module !AC\	
									43	41	21	20	2C	05	0049F	P.AAN:	.ASCII	<2>\, \	

KNOWN_EVENTS= P.AAA
KNOWN_COUNTERS= P.AAB

.EXTRN	EVL\$GL LOGMASK, EVL\$UNJULIAN
.EXTRN	WKQ\$ADD WORK_ITEM
.EXTRN	EVL\$NMA_LOS, EVL\$NMA_ANC
.EXTRN	EVL\$NMA_ALC, EVL\$NMA_ALS
.EXTRN	EVL\$NMA_LCZ, EVL\$NMA_NCZ
.EXTRN	EVL\$NMA_PSL, EVL\$NMA_ABS
.EXTRN	EVL\$NMA_CTR, EVL\$NMA_ZER
.EXTRN	EVL\$SCL_LNS, EVL\$SCL_ACR
.EXTRN	EVL\$NSL_IMS, EVL\$NSL_IFC


```
.EXTRN EVLS_NSL_DBR, EVLS_TPL_APL
.EXTRN EVLS_TPL_UPL, EVLS_TPL_RPL
.EXTRN EVLS_TPL_OPL, EVLS_TPL_PFM
.EXTRN EVLS_TPL_PRU, EVLS_TPL_VFR
.EXTRN EVLS_TPL_LDF, EVLS_TPL_CDS
.EXTRN EVLS_TPL_CDO, EVLS_TPL_LUP
.EXTRN EVLS_TPL_ILF, EVLS_TPL_ISF
.EXTRN EVLS_TPL_IOF, EVLS_TPL_RCH
.EXTRN EVLS_TPL_AUP, EVLS_TPL_ARJ
.EXTRN EVLS_TPL_ACH, EVLS_TPL_LDS
.EXTRN EVLS_TPL_LDO, EVLS_DLL_LSC
.EXTRN EVLS_DLL_RSC, EVLS_DLL_PRS
.EXTRN EVLS_DLL_SND, EVLS_DLL_RET
.EXTRN EVLS_DLL_SLC, EVLS_DLL_BHF
.EXTRN EVLS_DLL_SAD, EVLS_DLL_STT
.EXTRN EVLS_DLL_LBS, EVLS_DLL_RST
.EXTRN EVLS_DLL_STC, EVLS_DLL_RME
.EXTRN EVLS_DLL_IFL, EVLS_DLL_SFL
.EXTRN EVLS_DLL_RFL, EVLS_DLL_CDC
.EXTRN EVLS_DLL_DTU, EVLS_DLL_DTD
.EXTRN EVLS_PLL_DSR, EVLS_PLL_RIN
.EXTRN EVLS_PLL_CAR, EVLS_PLL_MEM
.EXTRN EVLS_PLL_COM, EVLS_PLL_PFM
.EXTRN EVLS_VMS_DBC, EVLS_VMS_DPC
.EXTRN EVLS_VMS_DP2, EVLS_VMS_PCR
.EXTRN EVLS_VMS_PTR, EVLS_CTNOO_ZER
.EXTRN EVLS_CTNOO_BRC, EVLS_CTNOO_BSN
.EXTRN EVLS_CTNOO_MRC, EVLS_CTNOO_MSN
.EXTRN EVLS_CTNOO_CRC, EVLS_CTNOO_CSN
.EXTRN EVLS_CTNOO_RTO, EVLS_CTNOO_RSE
.EXTRN EVLS_CTNOO_MLL, EVLS_CTNOO_APL
.EXTRN EVLS_CTNOO_NUL, EVLS_CTNOO_NOL
.EXTRN EVLS_CTNOO_OPL, EVLS_CTNOO_PFE
.EXTRN EVLS_CTNOO_RUL, EVLS_CTNOO_VER
.EXTRN EVLS CTCIR_ZER, EVLS CTCIR_APR
.EXTRN EVLS CTCIR_DPS, EVLS CTCIR_ACL
.EXTRN EVLS CTCIR_CRL, EVLS CTCIR_TPR
.EXTRN EVLS CTCIR_TPS, EVLS CTCIR_TCL
.EXTRN EVLS CTCIR_LDN, EVLS CTCIR_IFL
.EXTRN EVLS CTCIR_BRC, EVLS CTCIR_BSN
.EXTRN EVLS CTCIR_MBY, EVLS CTCIR_DBR
.EXTRN EVLS CTCIR_DBS, EVLS CTCIR_DEI
.EXTRN EVLS CTCIR_DEO, EVLS CTCIR_RRT
.EXTRN EVLS CTCIR_LRT, EVLS CTCIR_RBE
.EXTRN EVLS CTCIR_LBE, EVLS CTCIR_SIE
.EXTRN EVLS CTCIR_SLT, EVLS CTCIR_UBU
.EXTRN EVLS CTCIR_RPE, EVLS CTCIR_LPE
.EXTRN EVLS CTCIR_LIR, EVLS CTCIR_RIR
.EXTRN EVLS CTCIR_NIR, EVLS CTCIR_MNE
.EXTRN EVLS CTCIR_ERI, EVLS CTCIR_ERO
.EXTRN EVLS CTCIR_RTO, EVLS CTCIR_LTO
.EXTRN EVLS CTCIR_BER, EVLS CTCIR_BEL
.EXTRN EVLS CTLIN_APR, EVLS CTLIN_DPS
.EXTRN EVLS CTLIN_ACL, EVLS CTLIN_TPR
.EXTRN EVLS CTLIN_TPS, EVLS CTLIN_TCL
.EXTRN EVLS CTLIN_LDN, EVLS CTLIN_IFL
.EXTRN EVLS CTLIN_BRC, EVLS CTLIN_BSN
```

```
.EXTRN EVLS-CTLIN-MBY, EVLS-CTLIN-DBR
.EXTRN EVLS-CTLIN-DBS, EVLS-CTLIN-MBL
.EXTRN EVLS-CTLIN-BID, EVLS-CTLIN-BSI
.EXTRN EVLS-CTLIN-BSM, EVLS-CTLIN-DEI
.EXTRN EVLS-CTLIN-DEO, EVLS-CTLIN-RRT
.EXTRN EVLS-CTLIN-LRT, EVLS-CTLIN-RBE
.EXTRN EVLS-CTLIN-LBE, EVLS-CTLIN-SIE
.EXTRN EVLS-CTLIN-SLT, EVLS-CTLIN-SFL
.EXTRN EVLS-CTLIN-CDC, EVLS-CTLIN-RFL
.EXTRN EVLS-CTLIN-UFQ, EVLS-CTLIN-OVR
.EXTRN EVLS-CTLIN-SBU, EVLS-CTLIN-UBU
.EXTRN EVLS-CTLIN-RPE, EVLS-CTLIN-LPE
.EXTRN EVLS-CTLIN-MBS, EVLS-CTLIN-MSN
.EXTRN EVLS-CTLIN-RME, EVLS-CTLIN-LCE
.EXTRN EVLS-CTLIN-MSE, EVLS-CTXP-ZER
.EXTRN EVLS-CTXP-BRC, EVLS-CTXP-BSN
.EXTRN EVLS-CTXP-BLR, EVLS-CTXP-BLS
.EXTRN EVLS-CTXP-CRC, EVLS-CTXP-CSN
.EXTRN EVLS-CTXP-FSR, EVLS-CTXP-FSS
.EXTRN EVLS-CTXP-MSA, EVLS-CTXP-MCA
.EXTRN EVLS-CTXP-RSE, EVLS-CTXP-LIR
.EXTRN EVLS-CTXP-RIR, EVLS-CTXP-NIR
.EXTRN EVLS-CTXP-RST, EVLS-CTXS-ZER
.EXTRN EVLS-CTXS-MCA, EVLS-CTXS-ICR
.EXTRN EVLS-CTXS-LLR, SYSS$GETMSG
```

.PSECT \$CODE\$,NOWRT,2

```
.ENTRY FORMAT EVENT, Save R2,R3,R4,R5,R6,R7,R8,R9,-, 0813
R10,R11
MOVAB P.AAC, R11
MOVAB -680(SP), SP
CLRL DESC
MOVAB BUFFER, DESC+4
MNEGL #1, SHORT_FORM
MOVL EVENT, R2
EXTZV #0, #6, 2(R2), -(SP)
EXTZV #6, #9, 2(R2), -(SP)
PUSHL R11
PUSHAB DESC
CALLS #4, APPEND_STRINGS
MOVAB KNOWN_EVENTS, R3
TSTL (PTR)
BEQL 3$
CMPW 2(R2), (PTR)
BNEQ 2$
MOVZBL #128, TEXT
MOVAB TEXTBUF, TEXT+4
MOVQ #1, -(SP)
PUSHAB TEXT
PUSHAB TEXT
PUSHL 2(PTR)
CALLS #5, SYSS$GETMSG
PUSHAB TEXT
PUSHAB P.AAD
PUSHAB DESC
CALLS #3, APPEND_STRINGS
```

OFFC 00000

```
5B 0000' CF 9E 00002
5E FD58 CE 9E 00007
F0 AD D4 0000C
F4 0098 CE 9E 0000F
5A 01 CE 00015
52 04 AC D0 00018
06 00 EF 0001C
7E 02 A2 06 EF 00022
7E 02 A2 09 06 EF 00022
5B DD 00028
F0 AD 9F 0002A
0000V CF 04 FB 0002D
53 FBDO CB 9E 00032
63 D5 00037 1$:
38 13 00039
63 02 A2 B1 0003B
2D 12 0003F
10 AE 80 8F 9A 00041
14 AE 18 AE 9E 00046
7E 01 7D 0004B
18 AE 9F 0004E
1C AE 9F 00051
02 A3 DD 00054
00000000G 00 05 FB 00057
10 AE 9F 0005E
15 AB 9F 00061
F0 AD 9F 00064
0000V CF 03 FB 00067
```

60

6F

70

6D

69

63

69

61

77

6F

63

72

65

20

72

69

65

74

20

75

53		05	11	0006C	28:	BRB	38	0868
		06	C0	0006E		ADDL2	#6, PTR	0859
57	OC	C4	11	00071	38:	BRB	18	
		AC	D0	00073		MOVL	OUTPUT_ROUTINE, R7	0881
	FO	5A	DD	00077		PUSHL	SHORT_FORM	
56	10	AD	9F	00079		PUSHAB	DESC	
		AC	D0	0007C		MOVL	OUTPUT_PARM, R6	
67		56	DD	00080		PUSHL	R6	
	FO	03	FB	00082		CALLS	#3, (R7)	
7E	OA	AD	D4	00085		CLRL	DESC	
0000V	CF	A2	3C	00088		MOVZWL	10(R2), -(SP)	0885
		01	FB	0008C		CALLS	#1, FORMAT_NODEADR	
		50	DC	00091		PUSHL	R0	
	1B	AB	9F	00093		PUSHAB	P.AAE	
0000V	FO	AD	9F	00096		PUSHAB	DESC	
		03	FB	00099		CALLS	#3, APPEND_STRINGS	
	OC	A2	95	0009E		TSTB	12(R2)	0887
		0E	13	000A1		BEQL	48	
	OC	A2	9F	000A3		PUSHAB	12(R2)	0889
	29	AB	9F	000A6		PUSHAB	P.AAF	
0000V	FO	AD	9F	000A9		PUSHAB	DESC	
		03	FB	000AC	48:	CALLS	#3, APPEND_STRINGS	
08	AE	04	A2	000B1		MOVZWL	4(R2), DAYS	0891
04	AE	06	A2	000B6		MOVZWL	6(R2), SECONDS	0892
	6E	08	A2	000BB		MOVZWL	8(R2), MSECS	0893
		F8	AD	000BF		PUSHAB	TIME	0895
		04	AE	000C2		PUSHAB	MSECS	
	OC	AE	9F	000C5		PUSHAB	SECONDS	
0000G	CF	14	AE	000C8		PUSHAB	DAYS	
		04	FB	000CB		CALLS	#4, EVL\$UNJULIAN	
		F8	AD	000D0		PUSHAB	TIME	0897
		30	AB	000D3		PUSHAB	P.AAG	
0000V	FO	AD	9F	000D6		PUSHAB	DESC	
		03	FB	000D9		CALLS	#3, APPEND_STRINGS	
		5A	DD	000DE		PUSHL	SHORT_FORM	
	FO	AD	9F	000E0		PUSHAB	DESC	
		56	DD	000E3		PUSHL	R6	
67		03	FB	000E5		CALLS	#3, (R7)	
	FO	AD	D4	000E8		CLRL	DESC	
		53	94	000EB		CLRB	COMMA_NEEDED	0900
	OC	A2	9A	000ED		MOVZBL	12(R2), R0	0902
OC	OD	A042	9E	000F1		MOVAB	13(R0)[R2], PTR	
	OC	BE	9A	000F7		MOVZBL	@PTR, R4	0904
	OC	AE	D6	000FB		INCL	PTR	
		54	91	000FE		CMPB	R4, #1	0907
01		1C	12	00101		BNEQ	58	
	OC	AE	DD	00103		PUSHL	PTR	0908
	36	AB	9F	00106		PUSHAB	P.AAH	
	FO	AD	9F	00109		PUSHAB	DESC	
0000V	CF	03	FB	0010C		CALLS	#3, APPEND_STRINGS	
		OC	BE	9A	00111	MOVZBL	@PTR, R0	0909
	OC	AE	D6	00115		INCL	PTR	
		50	C0	00118		ADDL2	R0, PTR	
OC	AE	01	90	0011C	58:	MOVB	#1, COMMA_NEEDED	0910
	53	54	91	0011F		CMPB	R4, #3	0913
03		1C	12	00122		BNEQ	68	
	OC	AE	DD	00124		PUSHL	PTR	0914

		3F	AB	9F	00127	PUSHAB	P.AAI		
		FO	AD	9F	0012A	PUSHAB	DESC		
0000V	CF		03	FB	0012D	CALLS	#3, APPEND_STRINGS		
	50	OC	BE	9A	00132	MOVZBL	@PTR, R0	0915	
		OC	AE	D6	00136	INCL	PTR		
	OC		50	C0	00139	ADDL2	R0, PTR		
	53		01	90	0013D	MOVB	#1, COMMA_NEEDED	0916	
			54	D5	00140	TSTL	R4	0919	
			3B	12	00142	BNEQ	8\$		
	7E	OC	BE	3C	00144	MOVZWL	@PTR, -(SP)	0920	
0000V	CF		01	FB	00148	CALLS	#1, FORMAT_NODEADR		
			50	DD	0014D	PUSHL	R0		
		4B	AB	9F	0014F	PUSHAB	P.AAJ		
		FO	AD	9F	00152	PUSHAB	DESC		
0000V	CF		03	FB	00155	CALLS	#3, APPEND_STRINGS		
	OC		02	C0	0015A	ADDL2	#2, PTR	0921	
		OC	BE	95	0015E	TSTB	@PTR	0922	
			0E	13	00161	BEQL	7\$		
		OC	AE	DD	00163	PUSHL	PTR	0924	
		54	AB	9F	00166	PUSHAB	P.AAK		
		FO	AD	9F	00169	PUSHAB	DESC		
0000V	CF		03	FB	0016C	CALLS	#3, APPEND_STRINGS		
	50	OC	BE	9A	00171	MOVZBL	@PTR, R0	0925	
		OC	AE	D6	00175	INCL	PTR		
	OC		50	C0	00178	ADDL2	R0, PTR		
	53		01	90	0017C	MOVB	#1, COMMA_NEEDED	0926	
	05		54	91	0017F	CMPB	R4, #5	0929	
			15	12	00182	BNEQ	9\$		
	7E	OC	BE	9A	00184	MOVZBL	@PTR, -(SP)	0930	
		5B	AB	9F	00188	PUSHAB	P.AAL		
		FO	AD	9F	0018B	PUSHAB	DESC		
0000V	CF		03	FB	0018E	CALLS	#3, APPEND_STRINGS		
		OC	AE	D6	00193	INCL	PTR	0931	
	53		01	90	00196	MOVB	#1, COMMA_NEEDED	0932	
	04		54	91	00199	CMPB	R4, #4	0935	
			1C	12	0019C	BNEQ	10\$		
		OC	AE	DD	0019E	PUSHL	PTR	0936	
		64	AB	9F	001A1	PUSHAB	P.AAM		
		FO	AD	9F	001A4	PUSHAB	DESC		
0000V	CF		03	FB	001A7	CALLS	#3, APPEND_STRINGS		
	50	OC	BE	9A	001AC	MOVZBL	@PTR, R0	0937	
		OC	AE	D6	001B0	INCL	PTR		
	OC		50	C0	001B3	ADDL2	R0, PTR		
	53		01	90	001B7	MOVB	#1, COMMA_NEEDED	0938	
	55	FO	AD	D0	001BA	MOVL	DESC, R5	0948	
	52	08	AC	C1	001BE	ADDL3	EVENT_LENGTH, R2, R8	0942	
	58	OC	AE	D1	001C3	CPL	PTR, R8		
			7A	1E	001C7	BGEQU	18\$		
	2D		55	D1	001C9	CPL	R5, #45	0948	
			0F	15	001CC	BLEQ	12\$		
			5A	DD	001CE	PUSHL	SHORT_FORM	0950	
		FO	AD	9F	001D0	PUSHAB	DESC		
			56	DD	001D3	PUSHL	R6		
	67		03	FB	001D5	CALLS	#3, (R7)		
		FO	AD	D4	001D8	CLRL	DESC		
			53	94	001DB	CLRB	COMMA_NEEDED	0952	
	59	FO	AD	D0	001DD	MOVL	DESC, -STARTLEN	0955	

58

75

61
6C

72

6F

64

65

74

65

61

69

20

73

72

	0D		53	E9	001E1	BLBC	COMMA_NEEDED, 13\$	0957
		6F	AB	9F	001E4	PUSHAB	P.AAN	0960
		FO	AD	9F	001E7	PUSHAB	DESC	
	000V	CF	02	FB	001EA	CALLS	#2, APPEND_STRINGS	
			53	94	001EF	CLRB	COMMA_NEEDED	0961
	54	58	OC	AE	C3 001F1	SUBL3	PTR, R8, R4	0975
			OC	BE	B5 001F6	TSTW	APTR	0964
				27	18 001F9	BGEQ	16\$	
				5A	D5 001FB	TSTL	SHORT_FORM	0967
				05	18 001FD	BGEQ	14\$	
	5A			59	D0 001FF	MOVL	STARTLEN, SHORT_FORM	0969
				02	11 00202	BRB	15\$	
				5A	D4 00204	CLRL	SHORT_FORM	0972
				5A	DD 00206	PUSHL	SHORT_FORM	
			FO	AD	9F 00208	PUSHAB	DESC	
				56	DD 0020B	PUSHL	R6	
	67			03	FB 0020D	CALLS	#3, (R7)	
			FO	AD	D4 00210	CLRL	DESC	
				54	DD 00213	PUSHL	R4	0975
			10	AE	9F 00215	PUSHAB	PTR	0974
			FO	AD	9F 00218	PUSHAB	DESC	
	0000V	CF		03	FB 0021B	CALLS	#3, FORMAT_COUNTER	
				13	11 00220	BRB	17\$	0964
				06	EF 00222	EXTZV	#6, #9, 2(R2), -(SP)	0980
				54	DD 00228	PUSHL	R4	0979
			14	AE	9F 0022A	PUSHAB	PTR	0978
			FO	AD	9F 0022D	PUSHAB	DESC	
	0000V	CF		04	FB 00230	CALLS	#4, FORMAT_PARAMETER	
			FO	AD	D0 00235	MOVL	DESC, R5	0982
		55		59	D1 00239	CMPL	STARTLEN, R5	
				85	13 0023C	BEQL	11\$	
				01	90 0023E	MOVB	#1, COMMA_NEEDED	0984
		53		80	11 00241	BRB	11\$	0942
			FO	AD	D5 00243	TSTL	DESC	0987
				0D	15 00246	BLEQ	19\$	
				5A	DD 00248	PUSHL	SHORT_FORM	0988
			FO	AD	9F 0024A	PUSHAB	DESC	
				56	DD 0024D	PUSHL	R6	
	67			03	FB 0024F	CALLS	#3, (R7)	
			FO	AD	D4 00252	CLRL	DESC	
				5A	DD 00255	PUSHL	SHORT_FORM	0989
			FO	AD	9F 00257	PUSHAB	DESC	
				56	DD 0025A	PUSHL	R6	
				03	FB 0025C	CALLS	#3, (R7)	
	67		FO	AD	D4 0025F	CLRL	DESC	
				04	00262	RET		0993

; Routine Size: 611 bytes, Routine Base: \$CODE\$ + 0000

```

589 0994 1 ROUTINE append_strings (result, string, args): NOVALUE =
590 0995 1
591 0996 1 -----
592 0997 1
593 0998 1 Functional description
594 0999 1
595 1000 1 This routine appends an FAO result string to another
596 1001 1 string. The second argument is used as an FAO control
597 1002 1 string with the following arguments used as parameters.
598 1003 1 The result of the FAO operation is appended to the end
599 1004 1 of the string described by the first operand and the
600 1005 1 descriptor is updated.
601 1006 1
602 1007 1 Input parameters
603 1008 1
604 1009 1 result = Address of descriptor of string to be appended
605 1010 1 string = FAO control string (counted string)
606 1011 1 args = 0 to n FAO arguments
607 1012 1
608 1013 1 Output parameters
609 1014 1
610 1015 1 result = Updated descriptor of appended string
611 1016 1
612 1017 1 -----
613 1018 1
614 1019 2 BEGIN
615 1020 2
616 1021 2 MAP
617 1022 2 string: REF VECTOR [,BYTE], ! Address the counted string
618 1023 2 result: REF VECTOR; ! Address the descriptor
619 1024 2
620 1025 2 LOCAL
621 1026 2 faodesc: VECTOR [2], ! FAO control string descriptor
622 1027 2 desc: VECTOR [2]; ! FAO result string descriptor
623 1028 2
624 1029 2 faodesc [0] = .string [0]; ! Get length of counted string
625 1030 2 faodesc [1] = string [1];
626 1031 2
627 1032 2 desc [0] = max_output - .result [0]; ! Setup descriptor for FAO
628 1033 2 desc [1] = .result [1] + .result [0];
629 1034 2
630 P 1035 2 $FAOL(CTRSTR=faodesc, ! Insert string into buffer
631 P 1036 2 OUTLEN=desc,
632 P 1037 2 OUTBUF=desc,
633 1038 2 PRMLST=args);
634 1039 2
635 1040 2 result [0] = .result [0] + .desc [0]; ! Update string length
636 1041 2
637 1042 1 END;

```

.EXTRN SYSS\$FAOL

0004 00000 APPEND_STRINGS:

5E

OC C2 00002

.WORD Save R2
SUBL2 #12, SP

: 0994
:

CONSOLE
V04-000

D 15
16-Sep-1984 01:31:10
14-Sep-1984 12:28:46

VAX-11 BLISS-32 V4.0-742
DISK\$VMSMASTER:[EVL.SRC]CONSOLE.B32;1

Page 25
(4)

08	AE	04	AE	08	BC	9A	00005	MOVZBL	@STRING, FAODESC	:	1029
		08	AC		01	C1	0000A	ADDL3	#1, STRING, FAODESC+4	:	1030
			52	04	AC	D0	00010	MOVL	RESULT, R2	:	1032
	7E	00000200	8F		62	C3	00014	SUBL3	(R2), #512, DESC	:	
04	AE	04	A2		62	C1	0001C	ADDL3	(R2), 4(R2), DESC+4	:	1033
				0C	AC	9F	00022	PUSHAB	ARGS	:	1038
				04	AE	9F	00025	PUSHAB	DESC	:	
				08	AE	9F	00028	PUSHAB	DESC	:	
				14	AE	9F	0002B	PUSHAB	FAODESC	:	
	00000000G	00			04	FB	0002E	CALLS	#4, SYS\$FAOL	:	
		62			6E	C0	00035	ADDL2	DESC, (R2)	:	1040
					04		00038	RET		:	1042

; Routine Size: 57 bytes, Routine Base: \$CODE\$ + 0263

CON
V04

3D

20

61

69

65

21

6E

6C

20

67

65

72

20

6E

```

639 1043 1 ROUTINE format_parameter (string, param, maxlen, class): NOVALUE =
640 1044 1
641 1045 1 ---
642 1046 1
643 1047 1     Format a NICE parameter and append the
644 1048 1     descriptive text to a given string.
645 1049 1
646 1050 1     Inputs:
647 1051 1
648 1052 1     param = Address of longword containing pointer to parameter
649 1053 1     string = Address of descriptor of existing string
650 1054 1     maxlen = Maximum length that the parameter may be
651 1055 1     class = Event class
652 1056 1
653 1057 1     Outputs:
654 1058 1
655 1059 1     ptr = Address of longword pointing to location following parameter
656 1060 1 ---
657 1061 1
658 1062 2 BEGIN
659 1063 2
660 1064 2 BIND
661 1065 2     desc = .string: VECTOR;      ! Address of string descriptor
662 1066 2
663 1067 2 LOCAL
664 1068 2     ptr:          REF $BBLOCK,      ! Address of parameter
665 1069 2     param_type,    ! Parameter type (specific to class)
666 1070 2     nextptr:      REF $BBLOCK,      ! Pointer to next parameter
667 1071 2     fields,        ! # fields in parameter
668 1072 2     byte1:         BYTE,             ! First data byte after data type
669 1073 2     byte2:         SIGNED BYTE;      ! Second data byte after data type
670 1074 2
671 1075 2 MACRO
672 1076 2     choose(index) =
673 1077 2         CASE index FROM 0 TO %LENGTH-2
674 1078 2         OF
675 1079 2             SET
676 1080 2             choose_item(%REMAINING);
677 1081 2             TEST
678 1082 2             choose_item [string] =
679 1083 2             [%COUNT]: append('!AC',UPLIT BYTE(%ASCII string))%;
680 1084 2
681 1085 2     ptr = ..param;      ! Get address of parameter
682 1086 2     param_type = .ptr [0,0,11,0]; ! Get parameter type code
683 1087 2     nextptr = .ptr+2;    ! Assume first field follows
684 1088 2     IF .nextptr [0,7,1,0] ! If coded
685 1089 2         AND .nextptr [0,6,1,0] ! and multiple number of fields.
686 1090 2     THEN
687 1091 2         BEGIN
688 1092 2             fields = .nextptr [0,0,6,0]; ! Get # fields in parameter
689 1093 2             nextptr = .nextptr + 1;      ! and skip to actual first data item
690 1094 2         END
691 1095 2     ELSE
692 1096 2         fields = 1;      ! Else, indicate only one field
693 1097 2
694 1098 2     INCR i FROM 1 TO .fields ! For each field
695 1099 2     DO

```

```
696      1100      2      IF .nextptr [0,7,1,0]      ! If coded
697      1101      THEN
698      1102      nextptr = .nextptr + 1 + .nextptr [0,0,6,0] ! Use # coded bytes
699      1103      ELSE
700      1104      IF .nextptr [0,6,1,0]      ! If ASCII image field
701      1105      OR .nextptr [0,0,4,0] EQL 0 ! If zero, hex image field
702      1106      THEN
703      1107      nextptr = .nextptr + 2 + CH$RCHAR(.nextptr+1) ! Use ASCII length
704      1108      ELSE
705      1109      nextptr = .nextptr + 1 + .nextptr [0,0,4,0]; ! Else, use binary data length
706      1110
707      1111      byte1 = CH$RCHAR(.ptr+3);      ! Get first data byte
708      1112      byte2 = CH$RCHAR(.ptr+4);      ! and second data byte
709      1113      SELECT ONEU .class      ! For each layer,
710      1114      OF
711      1115      SET
712      1116
713      1117      [levc$c_cls_nma]:      ! Network management layer
714      1118      SELECT ONEU .param_type ! Case on parameter type
715      1119      OF
716      1120      SET
717      1121      [levc$c_nma_pser]: choose(.byte1, 'Load', 'Dump');
718      1122      [levc$c_nma_psts]:
719      1123      BEGIN
720      1124      choose(IF .byte2 EQL 0 THEN 0
721      1125      ELSE IF .byte2 LSS 0 THEN 1 ELSE 2,
722      1126      'Requested', '', 'Successful');
723      1127
724      1128      IF .byte2 LSS 0
725      1129      THEN
726      1130      choose(-.byte2-1,
727      1131      'Unrecognized function or option',
728      1132      'Invalid message format',
729      1133      'Privilege violation',
730      1134      'Oversized command message',
731      1135      'Management program error',
732      1136      'Unrecognized parameter type',
733      1137      'Incompatible management version',
734      1138      'Unrecognized component',
735      1139      'Invalid identification',
736      1140      'Line communication error',
737      1141      'Component in wrong state',
738      1142      'File open error',
739      1143      'Invalid file contents',
740      1144      'Resource error',
741      1145      'Invalid parameter value',
742      1146      'Line protocol error',
743      1147      'File I/O error',
744      1148      'Mirror link disconnected',
745      1149      'No room for new entry',
746      1150      'Mirror connect failed',
747      1151      'Parameter not applicable',
748      1152      'Parameter value too long',
749      1153      'Hardware failure',
750      1154      'Operation failure',
751      1155      'Function not supported',
752      1156      'Invalid parameter grouping',
```

```

'Bad loopback response',
'Parameter missing');
IF .fields GEQ 2      ! If NICE detail specified,
THEN                ! Then for certain errors,
  SELECTONE .byte2
  OF
    SET
    [nma$c_sts_fop]: (append(' ');
    choose(CH$RCHAR(.ptr+6), ! Expand on FOP detail
    'Permanent database',
    'Load file',
    'Dump file',
    'Secondary loader',
    'Tertiary loader',
    'Secondary dumper'));
    [nma$c_sts_cmp,nma$c_sts_ide,nma$c_sts_sta]:
    (append(' ');
    choose(CH$RCHAR(.ptr+6), ! Show entity name
    'Node',
    'Line',
    'Logging',
    'Circuit',
    'Module',
    'Area'));
    [nma$c_sts_pty,nma$c_sts_pva,nma$c_sts_pms]:
    (append(' ');
    CASE .(.ptr+6)<0,16> ! Show NODE parameter
    FROM nma$c_pcno_loa TO nma$c_pcno_tlo
    OF
      SET
      [nma$c_pcno_loa]: append('Load file');
      [nma$c_pcno_slo]: append('Secondary loader');
      [nma$c_pcno_tlo]: append('Tertiary loader');
      TES);
    TES;
  IF .fields GEQ 3      ! If optional message text,
  AND CH$RCHAR(.ptr+9) NEQ 0 ! and its non-null,
  THEN
    append('!/:!AC',.ptr+9); ! print it
  END;
[evc$c_nma_popr]: (append('Operation ');
choose(.byte1, 'Initiated', 'Terminated'));
[evc$c_nma_prsn]:
choose(.byte1,
'Receive timeout',
'Receive error',
'Line state change by higher level',
'Unrecognized request',
'Line open error');
[evc$c_nma_pnod]: (append('Node = !AS'
format_nodeadr(.ptr [4,0,16,0]));
IF .fields GTR 1
AND CH$RCHAR(.ptr+7) NEQ 0
THEN append(' (!AC',.ptr+7));
[evc$c_nma_pdte]: append('DTE = !AC',.ptr+3);
[evc$c_nma_pfil]: append('File = !AC',.ptr+3);

```

```

753 P 1157
754 1158
755 1159
756 1160
757 1161
758 1162
759 1163
760 1164
761 P 1165
762 P 1166
763 P 1167
764 P 1168
765 P 1169
766 P 1170
767 1171
768 1172
769 1173
770 P 1174
771 P 1175
772 P 1176
773 P 1177
774 P 1178
775 P 1179
776 1180
777 1181
778 1182
779 1183
780 1184
781 1185
782 1186
783 1187
784 1188
785 1189
786 1190
787 1191
788 1192
789 1193
790 1194
791 1195
792 1196
793 1197
794 1198
795 1199
796 1200
797 P 1201
798 P 1202
799 P 1203
800 P 1204
801 P 1205
802 1206
803 P 1207
804 1208
805 1209
806 1210
807 1211
808 1212
809 1213

```



```

810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866

```

P

P

P

```

[evc$sc_nma_psty]:
    choose(.byte1,
        'Secondary loader',
        'Tertiary loader',
        'Operating system');

[evc$sc_nma_psn]:      ! Target's Ethernet address
    BEGIN
    append('Ethernet address = !XB', .(.ptr+4)<0,8>);
    INCR p FROM 1 TO 5 DO
        append('!XB', .(.ptr+4+.p)<0,8>);
    END;

[9]: append('DTE = !AC', .ptr+3);    !! 88 temporary until PSI gets fixed

[OTHERWISE]: unknown_parameter(desc, .ptr, .maxlen);
TES;

[evc$sc_cls_scl]:      ! Session control layer
    SELECT .param_type
    OF
        SET
        [evc$sc_scl_psrn]:
            choose(.byte1, 'Operator command', 'Normal operation');
        [evc$sc_scl_pold]: append('Old state = ');
        [evc$sc_scl_pnew]: append('New state = ');
        [evc$sc_scl_pold, evc$sc_scl_pnew]:
            choose(.byte1, 'On', 'Off', 'Shut', 'Restricted');
        [evc$sc_scl_pnod]: (append('Source node = !AS',
            format_nodeadr(.ptr [4,0,16,0])));
        IF .fields GTR 1
            AND CH$RCHAR(.ptr+7) NEQ 0
            THEN append('(!AC', .ptr+7));
        [evc$sc_scl_pspc]: append('Source process = ');
        [evc$sc_scl_pdpc]: append('Destination process = ');
        [evc$sc_scl_pspc, evc$sc_scl_pdpc]:
            (append('!UL', CH$RCHAR(.ptr+4)));
        IF .fields GTR 3
            THEN append('(!AC', .ptr+10));
        [evc$sc_scl_pusr]: append('User = !AC', .ptr+3);
        [evc$sc_scl_ppsw]: append('Password');
        [evc$sc_scl_pacc]: append('Account = !AC', .ptr+3);
        [OTHERWISE]: unknown_parameter(desc, .ptr, .maxlen);
        TES;

[evc$sc_cls_nsl]:      ! Network services layer
    SELECT ONEU .param_type
    OF
        SET
        [evc$sc_nsl_pmsg]:
            (append('Message = ');
            unknown_param_data(desc, .ptr+2, .maxlen));
        [evc$sc_nsl_pflo]: append('Current flow control = !SB', .byte1);
        [evc$sc_nsl_pnod]: (append('Source node = !AS',
            format_nodeadr(.ptr [4,0,16,0])));
        IF .fields GTR 1
            AND CH$RCHAR(.ptr+7) NEQ 0

```

867 1271
868 1272
869 1273
870 1274
871 1275
872 1276
873 1277
874 1278
875 1279
876 1280
877 1281
878 1282
879 1283
880 1284
881 1285
882 1286
883 1287
884 1288
885 1289
886 1290
887 1291
888 1292
889 1293
890 1294
891 1295
892 1296
893 1297
894 1298
895 1299
896 1300
897 1301
898 1302
899 1303
900 1304
901 1305
902 1306
903 1307
904 1308
905 1309
906 1310
907 1311
908 1312
909 1313
910 1314
911 1315
912 1316
913 1317
914 1318
915 1319
916 1320
917 1321
918 1322
919 1323
920 1324
921 1325
922 1326
923 P 1327

```

        THEN append(' (!AC)', .ptr+7));
[OTHERWISE]: unknown_parameter(desc, .ptr, .maxlen);
TES;

[evc$c_cls_tpl]:      ! Transport Layer
  SE[ECTO] .param_type
  OF
    SET
    [evc$c_tpl_ppkh]:
      (append('Packet header = ');
      unknown_param_data(desc, .ptr+2, .maxlen));
    [evc$c_tpl_ppkb]:
      (append('Packet beginning = ');
      unknown_param_data(desc, .ptr+2, .maxlen));
    [evc$c_tpl_phia]: append('Highest address = !UL', .ptr [3,0,16,0]);
    [evc$c_tpl_pnod]: append('Node = ');
    [evc$c_tpl_padj]: append('Adjacent node = ');
    [evc$c_tpl_pexp]: append('Expected node = ');
    [evc$c_tpl_pnod, evc$c_tpl_pexp, evc$c_tpl_padj]:
      (append('!AS', format_nodeadr(.ptr [4,0,16,0]));
      IF .fields GTR 1
      AND CH$RCHAR(.ptr+7) NEQ 0
      THEN append(' (!AC)', .ptr+7));
    [evc$c_tpl_prsn]:
      choose(.byte1,
        'Line synchronization lost',
        'Data errors',
        'Unexpected packet type',
        'Routing update checksum error',
        'Adjacent node address change',
        'Verification receive timeout',
        'Version skew',
        'Adjacent node address out of range',
        'Adjacent node block size too small',
        'Invalid verification seed value',
        'Adjacent node listener receive timeout',
        'Adjacent node listener received invalid data',
        'Call failed',
        'Verification password required from Phase III node',
        'Dropped by adjacent node');
    [evc$c_tpl_pvr]: append('Received version = !UL.!UL.!UL',
      CH$RCHAR(.ptr+4),
      CH$RCHAR(.ptr+6),
      CH$RCHAR(.ptr+8));
    [evc$c_tpl_psts]: (
      choose(.byte1, 'Reachable', 'Unreachable'));
[OTHERWISE]: unknown_parameter(desc, .ptr, .maxlen);
TES;

[evc$c_cls_dll]:      ! Data Link Layer
  SE[ECTO] .param_type
  OF
    SET
    [evc$c_dll_pold]: append('Old state = ');
    [evc$c_dll_pnew]: append('New state = ');
    [evc$c_dll_pold, evc$c_dll_pnew]:
      choose(.byte1, 'Halted', 'ISTRT', 'ASTRT', 'Running',

```

```

924 1328
925 1329
926 1330
927 1331
928 1332
929 1333
930 1334
931 1335
932 1336
933 1337
934 1338
935 1339
936 1340
937 1341
938 1342
939 1343
940 1344
941 1345
942 1346
943 1347
944 1348
945 1349
946 1350
947 1351
948 1352
949 1353
950 1354
951 1355
952 1356
953 1357
954 1358
955 1359
956 1360
957 1361
958 1362
959 1363
960 1364
961 1365
962 1366
963 1367
964 1368
965 1369
966 1370
967 1371
968 1372
969 1373
970 1374
971 1375
972 1376
973 1377
974 1378
975 1379
976 1380
977 1381
978 1382
979 1383
980 1384

```

```

'Maintenance');
[evc$dll_phdr]:
  (append('Header = ');
  unknown_param_data(desc, .ptr+2, .maxlen));
[evc$dll_pslt]: append('Selected tributary = !UL', .byte1);
[evc$dll_ppvt]: append('Previous tributary = !UL', .byte1);
[evc$dll_ptst]: (append('Tributary status = ');
  choose(.byte1,
    'Streaming',
    'Continued send after timeout',
    'Continued send after deselect',
    'Ended streaming'));
[evc$dll_prtb]: append('Received tributary = !UL', .byte1);
[evc$dll_pblk]: append('Block length = !UL', .ptr [3,0,16,0]);
[evc$dll_pblf]: append('Buffer length = !UL', .ptr [3,0,16,0]);
[evc$dll_pdte]: append('DTE = !AC', .ptr+3);
[evc$dll_prsn]: (append('Reason = ');
  choose(.byte1,
    'Operator command',
    'Normal operation'));
[evc$dll_post]: append('Old state = ');
[evc$dll_pnst]: append('New state = ');
[evc$dll_post, evc$dll_pnst]:
  choose(.byte1, 'On', 'Off', 'Shut');
[evc$dll_ptyp]: append('Parameter type = #!UL', .ptr [3,0,16,0]);
[evc$dll_pcau]: append('Cause = !UL', .byte1);
[evc$dll_pdia]: append('Diagnostic = !UL', .byte1);
[evc$dll_pfrs]: (append('Failure reason = ');
  choose(.byte1,
    'Excessive collisions',
    'Carrier check failed',
    'Short circuit',
    'Open circuit',
    'Frame too long',
    'Remote failure to defer',
    'Block check error',
    'Framing error',
    'Data overrun',
    'System buffer unavailable',
    'User buffer unavailable',
    'Unrecognized frame destination'));
[evc$dll_pdis]: append('Distance = !UL', .ptr [3,0,16,0]);
[evc$dll_pehd]: (append('Ethernet header = ');
  unknown_param_data(desc, .ptr+2, .maxlen));
[evc$dll_phws]: (append('Hardware status = ');
  unknown_param_data(desc, .ptr+2, .maxlen));
[OTHERWISE]: unknown_parameter(desc, .ptr, .maxlen);
TES;

[evc$cls_pll]: ! Physical Link Layer
SELECT .param_type
OF
SET
[evc$pll_pdvr]: append('Device register = !XW', .ptr [3,0,16,0]);
[evc$pll_pnew]: (append('New state = ');

```

```
981      choose(.byte1,'Off','On'));
982      [OTHERWISE]: unknown_parameter(desc, .ptr, .maxlen);
983      TES;
984
985      [69,225,226]:      ! DECnet/SNA Gateway (69) specific events
986                        ! and Server Base (225) specific events
987                        ! and Router Server (226) specific events
988                        ! (All parameters are text)
989      unknown_param_data(desc, .ptr+2, .maxlen);
990
991      [evc$c_cls_vms]:      ! VMS specific events
992      SELECT .param_type
993      OF
994      SET
995      [evc$c_vms_pnod]: (append('Remote node ');
996                        IF CH$RCHAR(.ptr+3) EQL 0      ! 0=address, else name
997                        THEN
998                          append('!AS', format_nodeadr(.ptr [4,0,16,0]));
999                        ELSE
1000                          append('!AC', .ptr+3));
1001      [evc$c_vms_pprc]: append('Name = !AC', .ptr+3);
1002      [evc$c_vms_ppid]: append('PID = !XL', .ptr [3,0,32,0]);
1003      [evc$c_vms_psts]: append('Status = !XL', .ptr [3,0,32,0]);
1004      [OTHERWISE]: unknown_parameter(desc, .ptr, .maxlen);
1005      TES;
1006
1007      [OTHERWISE]: unknown_parameter(desc, .ptr, .maxlen);
1008      TES;
1009
1010      .param = .nextptr;      ! Skip to next parameter
1011
1012      1 END;
```

										.PSECT		\$SPLITS, NOWRT, NOEXE, 2												
										64	43	41	21	03	004A2	P.AAO:	.ASCII	<3>\!AC\						
											61	6F	4C	04	004A6	P.AAP:	.ASCII	<4>\Load\						
											43	41	21	03	004AB	P.AAQ:	.ASCII	<3>\!AC\						
										70	6D	75	44	04	004AF	P.AAR:	.ASCII	<4>\Dump\						
											43	41	21	03	004B4	P.AAS:	.ASCII	<3>\!AC\						
										64	65	74	73	65	75	71	65	52	09	004B8	P.AAT:	.ASCII	<9>\Requested\	
											43	41	21	03	004C2	P.AAU:	.ASCII	<3>\!AC\						
														00	004C6	P.AAV:	.ASCII	<0>						
											43	41	21	03	004C7	P.AAW:	.ASCII	<3>\!AC\						
										6C	75	66	73	73	65	63	63	75	53	0A	004CB	P.AAX:	.ASCII	<10>\Successful\
											43	41	21	03	004D6	P.AAY:	.ASCII	<3>\!AC\						
66	20	64	65	7A	69	6E	67	6F	63	65	72	6E	55	1F	004DA	P.AAZ:	.ASCII	<31>\Unrecognized function or option\						
69	74	70	6F	20	72	6F	20	6E	6F	69	74	63	6E	75	004E9									
													6E	6F	004F8									
											43	41	21	03	004FA	P.ABA:	.ASCII	<3>\!AC\						
67	61	73	73	65	6D	20	64	69	6C	61	76	6E	49	16	004FE	P.ABB:	.ASCII	<22>\Invalid message format\						
											74	61	6D	72	6F	66	20	65	0050D					
											43	41	21	03	00515	P.ABC:	.ASCII	<3>\!AC\						
6C	6F	69	76	20	65	67	65	6C	69	76	69	72	50	13	00519	P.ABD:	.ASCII	<19>\Privilege violation\						
											6E	6F	69	74	61	00528								

6D	6D	6F	63	20	64	65	7A	69	73	72	43	41	21	03	0052D	P.ABE:	.ASCII	<3>\!AC\	
				65	67	61	73	73	65	6D	65	76	4F	19	00531	P.ABF:	.ASCII	<25>\Oversized command message\	
6F	72	70	20	74	6E	65	6D	65	67	61	6E	61	4D	18	00540				
				72	6F	72	72	65	20	6D	61	72	67	03	0054B	P.ABG:	.ASCII	<3>\!AC\	
														03	0054F	P.ABH:	.ASCII	<24>\Management program error\	
70	20	64	65	7A	69	6E	67	6F	63	65	72	6E	55	1B	00568	P.ABI:	.ASCII	<3>\!AC\	
		65	70	79	74	20	72	65	74	65	6D	61	72	61	0056C	P.ABJ:	.ASCII	<27>\Unrecognized parameter type\	
6D	20	65	6C	62	69	74	61	70	6D	6F	63	6E	49	1F	0057B				
69	73	72	65	76	20	74	6E	65	6D	65	67	61	6E	61	00588	P.ABK:	.ASCII	<3>\!AC\	
															0058C	P.ABL:	.ASCII	<31>\Incompatible management version\	
															0059B				
															005AA				
63	20	64	65	7A	69	6E	67	6F	63	65	43	41	21	03	005AC	P.ABM:	.ASCII	<3>\!AC\	
							74	6E	65	6E	6F	70	6D	6F	005B0	P.ABN:	.ASCII	<22>\Unrecognized component\	
69	74	6E	65	64	69	20	64	69	6C	61	76	6E	49	16	005BF				
							6E	6F	69	74	61	63	69	66	005C7	P.ABO:	.ASCII	<3>\!AC\	
															005CB	P.ABP:	.ASCII	<22>\Invalid identification\	
61	63	69	6E	75	6D	6D	6F	63	20	65	6E	69	4C	18	005DA	P.ABQ:	.ASCII	<3>\!AC\	
					72	6F	72	72	65	20	6E	6F	69	74	005E2	P.ABR:	.ASCII	<24>\Line communication error\	
															C05F5				
77	20	6E	69	20	74	6E	65	6E	6F	70	6D	6F	43	18	005FF	P.ABS:	.ASCII	<3>\!AC\	
					65	74	61	74	73	20	67	6E	6F	72	00603	P.ABT:	.ASCII	<24>\Component in wrong state\	
															00612				
															0061C	P.ABU:	.ASCII	<3>\!AC\	
															00620	P.ABV:	.ASCII	<0>	
6F	72	72	65	20	6E	65	70	6F	20	65	43	41	21	03	00621	P.ABW:	.ASCII	<3>\!AC\	
											6C	69	46	0F	00625	P.ABX:	.ASCII	<15>\File open error\	
															00634				
63	20	65	6C	69	66	20	64	69	6C	61	76	6E	49	15	00635	P.ABY:	.ASCII	<3>\!AC\	
							73	74	6E	65	74	6E	6F	03	00639	P.ABZ:	.ASCII	<21>\Invalid file contents\	
72	6F	72	72	65	20	65	63	72	75	6F	73	65	52	0E	00648	P.ACA:	.ASCII	<3>\!AC\	
															0064F	P.ACB:	.ASCII	<14>\Resource error\	
65	6D	61	72	61	70	20	64	69	6C	61	76	6E	49	17	00653	P.ACC:	.ASCII	<3>\!AC\	
					65	75	6C	61	76	20	72	65	74	03	00662	P.ACD:	.ASCII	<23>\Invalid parameter value\	
20	6C	6F	63	6F	74	6F	72	70	20	65	43	41	21	03	00666	P.ACE:	.ASCII	<3>\!AC\	
											6E	69	4C	13	00675	P.ACF:	.ASCII	<19>\Line protocol error\	
											72	6F	72	72	0067E				
72	6F	72	72	65	20	4F	2F	49	20	65	43	41	21	03	00691	P.ACG:	.ASCII	<3>\!AC\	
											6C	69	46	0E	00696	P.ACH:	.ASCII	<14>\File I/O error\	
69	64	20	6B	6E	69	6C	20	72	6F	72	43	41	21	03	0069A	P.ACI:	.ASCII	<3>\!AC\	
				64	65	74	63	65	6E	6E	6F	6F	63	73	006A9	P.ACJ:	.ASCII	<24>\Mirror link disconnected\	
65	6E	20	72	6F	66	20	6D	6F	6F	72	43	41	21	03	006AD				
											20	6F	4E	15	006BC	P.ACK:	.ASCII	<3>\!AC\	
											6E	65	20	77	006C6	P.ACL:	.ASCII	<21>\No room for new entry\	
74	63	65	6E	6E	6F	63	20	72	6F	72	43	41	21	03	006CA				
											72	69	4D	15	006D9	P.ACM:	.ASCII	<3>\!AC\	
											69	61	66	20	006E0	P.ACN:	.ASCII	<21>\Mirror connect failed\	
20	74	6F	6E	20	72	65	74	65	6D	61	43	41	21	03	006E4				
					65	6C	62	61	63	69	6C	70	70	61	006F3	P.ACO:	.ASCII	<3>\!AC\	
75	6C	61	76	20	72	65	74	65	6D	61	43	41	21	03	006FA	P.ACP:	.ASCII	<24>\Parameter not applicable\	
															0070D				
															00717	P.ACQ:	.ASCII	<3>\!AC\	
															0071B	P.ACR:	.ASCII	<24>\Parameter value too long\	

CONSOLE
V04-000

M 15
16-Sep-1984 01:31:10
14-Sep-1984 12:28:46

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EVL.SRC]CONSOLE.B32;1

Page 34
(5)

CONS
V04-

					67	6E	6F	6C	20	6F	6F	74	20	65	0072A						
											43	41	21	03	00734	P.ACS:	.ASCII	<3>\!AC\			
75	6C	69	61	66	20	65	72	61	77	64	72	61	48	10	00738	P.ACT:	.ASCII	<16>\Hardware failure\			
													65	72	00747						
											43	41	21	03	00749	P.ACU:	.ASCII	<3>\!AC\			
6C	69	61	66	20	6E	6F	69	74	61	72	65	70	4F	11	0074D	P.ACV:	.ASCII	<17>\Operation failure\			
												65	72	75	0075C						
											43	41	21	03	0075F	P.ACW:	.ASCII	<3>\!AC\			
73	20	74	6F	6E	20	6E	6F	69	74	63	6E	75	46	16	00763	P.ACX:	.ASCII	<22>\Function not supported\			
							64	65	74	72	6F	70	70	75	00772						
											43	41	21	03	0077A	P.ACY:	.ASCII	<3>\!AC\			
65	6D	61	72	61	70	20	64	69	6C	61	76	6E	49	1A	0077E	P.ACZ:	.ASCII	<26>\Invalid parameter grouping\			
			67	6E	69	70	75	6F	72	67	20	72	65	74	0078D						
											43	41	21	03	00799	P.ADA:	.ASCII	<3>\!AC\			
72	20	6B	63	61	62	70	6F	6F	6C	20	64	61	42	15	0079D	P.ADB:	.ASCII	<21>\Bad loopback response\			
								65	73	6E	6F	70	73	65	007AC						
											43	41	21	03	007B3	P.ADC:	.ASCII	<3>\!AC\			
73	73	69	6D	20	72	65	74	65	6D	61	72	61	50	11	007B7	P.ADD:	.ASCII	<17>\Parameter missing\			
												67	6E	69	007C6						
											43	41	21	03	007C9	P.ADE:	.ASCII	<2>\, \			
											43	41	21	03	007CC	P.ADF:	.ASCII	<3>\!AC\			
61	74	61	64	20	74	6E	65	6E	61	6D	72	65	50	12	007D0	P.ADG:	.ASCII	<18>\Permanent database\			
											65	73	61	62	007DF						
											43	41	21	03	007E3	P.ADH:	.ASCII	<3>\!AC\			
						65	6C	69	66	20	64	61	6F	4C	09	007E7	P.ADI:	.ASCII	<9>\Load file\		
											43	41	21	03	007F1	P.ADJ:	.ASCII	<3>\!AC\			
						65	6C	69	66	20	70	6D	75	44	09	007F5	P.ADK:	.ASCII	<9>\Dump file\		
											43	41	21	03	007FF	P.ADL:	.ASCII	<3>\!AC\			
64	61	6F	6C	20	79	72	61	64	6E	6F	63	65	53	10	00803	P.ADM:	.ASCII	<16>\Secondary loader\			
													72	65	00812						
											43	41	21	03	00814	P.ADN:	.ASCII	<3>\!AC\			
65	64	61	6F	6C	20	79	72	61	69	74	72	65	54	0F	00818	P.ADO:	.ASCII	<15>\Tertiary loader\			
													72	65	00827						
											43	41	21	03	00828	P.ADP:	.ASCII	<3>\!AC\			
70	6D	75	64	20	79	72	61	64	6E	6F	63	65	53	10	0082C	P.ADQ:	.ASCII	<16>\Secondary dumper\			
													72	65	0083B						
												20	2C	02	0083D	P.ADR:	.ASCII	<2>\, \			
											43	41	21	03	00840	P.ADS:	.ASCII	<3>\!AC\			
										65	64	6F	4E	04	00844	P.ADT:	.ASCII	<4>\Node\			
											43	41	21	03	00849	P.ADU:	.ASCII	<3>\!AC\			
										65	6E	69	4C	04	0084D	P.ADV:	.ASCII	<4>\Line\			
											43	41	21	03	00852	P.ADW:	.ASCII	<3>\!AC\			
						67	6E	69	67	67	6F	4C	07	07	00856	P.ADX:	.ASCII	<7>\Logging\			
											43	41	21	03	0085E	P.ADY:	.ASCII	<3>\!AC\			
						74	69	75	63	72	69	43	07	07	00862	P.ADZ:	.ASCII	<7>\Circuit\			
											43	41	21	03	0086A	P.AEA:	.ASCII	<3>\!AC\			
							65	6C	75	64	6F	4D	06	06	0086E	P.AEB:	.ASCII	<6>\Module\			
											43	41	21	03	00875	P.AEC:	.ASCII	<3>\!AC\			
										61	65	72	41	04	00879	P.AED:	.ASCII	<4>\Area\			
											20	2C	02	02	0087E	P.AEE:	.ASCII	<2>\, \			
64	61	6F	6C	20	79	72	61	64	6E	6F	63	65	53	10	00881	P.AEF:	.ASCII	<9>\Load file\			
													72	65	0088B	P.AEG:	.ASCII	<16>\Secondary loader\			
65	64	61	6F	6C	20	79	72	61	69	74	72	65	54	0F	0089A						
													72	65	0089C	P.AEH:	.ASCII	<15>\Tertiary loader\			
													72	65	008AB						
											43	41	21	2F	21	05	008AC	P.AEI:	.ASCII	<5>\!//!AC\	
					20	6E	6F	69	74	61	72	65	70	4F	0A	008B2	P.AFJ:	.ASCII	<10>\Operation \		

CONSOLE
V04-000

N 15
16-Sep-1984 01:31:10
14-Sep-1984 12:28:46

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EVL.SRC]CONSOLE.B32;1

Page 35
(5)

75	6F	65	6D	69	74	20	65	76	69	65	63	65	52	0F	008BD	P.AEK:	.ASCII	<3>\!AC\
															008C1	P.AEL:	.ASCII	<9>\Initiated\
															008CB	P.AEM:	.ASCII	<3>\!AC\
															008CF	P.AEN:	.ASCII	<10>\Terminated\
															008DA	P.AEO:	.ASCII	<3>\!AC\
															008DE	P.AEP:	.ASCII	<15>\Receive timeout\
															008ED			
															008EE	P.AEQ:	.ASCII	<3>\!AC\
															008F2	P.AER:	.ASCII	<13>\Receive error\
61	68	63	20	65	74	61	74	73	20	65	6E	69	4C	21	00900	P.AES:	.ASCII	<3>\!AC\
6C	20	72	65	68	67	69	68	20	79	62	20	65	67	6E	00904	P.AET:	.ASCII	\!Line state change by higher level\
															00913			
															00922			
72	20	64	65	7A	69	6E	67	6F	63	65	72	6E	55	14	00926	P.AEU:	.ASCII	<3>\!AC\
															0092A	P.AEV:	.ASCII	<20>\Unrecognized request\
															00939			
6F	72	72	65	20	6E	65	70	6F	20	65	6E	69	4C	0F	0093F	P.AEW:	.ASCII	<3>\!AC\
															00943	P.AEX:	.ASCII	<15>\Line open error\
															00952			
															00953	P.AEY:	.ASCII	<10>\Node = !AS\
															0095E	P.AEZ:	.ASCII	<6>\ (!AC)\
															00965	P.AFA:	.ASCII	<9>\DTE = !AC\
															0096F	P.AFB:	.ASCII	<10>\File = !AC\
64	61	6F	6C	20	79	72	61	64	6E	6F	63	65	53	10	0097A	P.AFC:	.ASCII	<3>\!AC\
															0097E	P.AFD:	.ASCII	<16>\Secondary loader\
															0098D			
65	64	61	6F	6C	20	79	72	61	69	74	72	65	54	0F	0098F	P.AFE:	.ASCII	<3>\!AC\
															00993	P.AFF:	.ASCII	<15>\Tertiary loader\
															009A2			
74	73	79	73	20	67	6E	69	74	61	72	65	70	4F	10	009A3	P.AFG:	.ASCII	<3>\!AC\
															009A7	P.AFH:	.ASCII	<16>\Operating system\
															009B6			
65	72	64	64	61	20	74	65	6E	72	65	68	74	45	16	009B8	P.AFI:	.ASCII	<22>\Ethernet address = !XB\
															009C7			
															009CF	P.AFJ:	.ASCII	<4>\-!XB\
															009D4	P.AFK:	.ASCII	<3>\!AC\
61	6D	6D	6F	63	20	72	6F	74	61	72	65	70	4F	10	009D8	P.AFL:	.ASCII	<16>\Operator command\
															009E7			
69	74	61	72	65	70	6F	20	6C	61	6D	72	6F	4E	10	009E9	P.AFM:	.ASCII	<3>\!AC\
															009ED	P.AFN:	.ASCII	<16>\Normal operation\
															009FC			
															009FE	P.AFO:	.ASCII	<12>\Old state = \
															00A0B	P.AFP:	.ASCII	<12>\New state = \
															00A18	P.AFQ:	.ASCII	<3>\!AC\
															00A1C	P.AFR:	.ASCII	<2>\On\
															00A1F	P.AFS:	.ASCII	<3>\!AC\
															00A23	P.AFT:	.ASCII	<3>\Off\
															00A27	P.AFU:	.ASCII	<3>\!AC\
															00A2B	P.AFV:	.ASCII	<4>\Shut\
															00A30	P.AFW:	.ASCII	<3>\!AC\
20	3D	20	65	64	6F	6E	20	65	63	72	75	6F	53	11	00A34	P.AFX:	.ASCII	<10>\Restricted\
															00A3F	P.AFY:	.ASCII	<17>\Source node = !AS\
															00A4E			
73	73	65	63	6F	72	70	20	65	63	72	75	6F	53	11	00A51	P.AFZ:	.ASCII	<6>\ (!AC)\
															00A58	P.AGA:	.ASCII	<17>\Source process = \
															00A67			
72	70	20	6E	6F	69	74	61	6E	69	74	73	65	44	16	00A6A	P.AGB:	.ASCII	<22>\Destination process = \

20	65	64	6F	6E	20	74	6E	65	63	61	6A	64	41	26	00C91	P.AHO:	.ASCII	\&Adjacent node listener receive timeout\	:	
76	69	65	63	65	72	20	72	65	6E	65	74	73	69	6C	00CA0				:	
						74	75	6F	65	6D	69	74	20	65	00CAF				:	
											43	41	21	03	00CB8	P.AHP:	.ASCII	<3>\!AC\	:	
20	65	64	6F	6E	20	74	6E	65	63	61	6A	64	41	2C	00CBC	P.AHQ:	.ASCII	\,Adjacent node listener received invalid\	:	
76	69	65	63	65	72	20	72	65	6E	65	74	73	69	6C	00CCB				:	
					64	69	6C	61	76	6E	69	20	64	65	00CDA				:	
										61	74	61	64	20	00CE4		.ASCII	\ data\	:	
											43	41	21	03	00CE9	P.AHR:	.ASCII	<3>\!AC\	:	
											43	41	21	03	00CED	P.AHS:	.ASCII	<11>\Call failed\	:	
											43	41	21	03	00CF9	P.AHT:	.ASCII	<3>\!AC\	:	
70	20	6E	6F	69	74	61	63	69	66	69	72	65	56	32	00CFD	P.AHU:	.ASCII	\2Verification password required from Pha\	:	
65	72	69	75	71	65	72	20	64	72	6F	77	73	73	61	00D0C				:	
						61	68	50	20	6D	6F	72	66	20	64	00D1B			:	
						65	64	6F	6E	20	49	49	49	20	65	73	00D25	.ASCII	\se III node\	:
											43	41	21	03	00D30	P.AHV:	.ASCII	<3>\!AC\	:	
6A	64	61	20	79	62	20	64	65	70	70	6F	72	44	18	00D34	P.AHW:	.ASCII	<24>\Dropped by adjacent node\	:	
											6E	65	63	61	00D43				:	
69	73	72	65	76	20	64	65	76	69	65	63	65	52	1E	00D4D	P.AHX:	.ASCII	<30>\Received version = !UL.!UL.!UL\	:	
55	21	2E	4C	55	21	2E	4C	55	21	20	3D	20	6E	6F	00D5C				:	
														4C	00D6B				:	
											43	41	21	03	00D6C	P.AHY:	.ASCII	<3>\!AC\	:	
											43	41	21	03	00D70	P.AHZ:	.ASCII	<9>\Reachable\	:	
											43	41	21	03	00D7A	P.AIA:	.ASCII	<3>\!AC\	:	
											43	41	21	03	00D7E	P.AIB:	.ASCII	<11>\Unreachable\	:	
											43	41	21	03	00D8A	P.AIC:	.ASCII	<12>\Old state = \	:	
											43	41	21	03	00D97	P.AID:	.ASCII	<12>\New state = \	:	
											43	41	21	03	00DA4	P.AIE:	.ASCII	<3>\!AC\	:	
											43	41	21	03	00DA8	P.AIF:	.ASCII	<6>\Halted\	:	
											43	41	21	03	00DAF	P.AIG:	.ASCII	<3>\!AC\	:	
											43	41	21	03	00DB3	P.AIH:	.ASCII	<5>\!ISTR\	:	
											43	41	21	03	00DB9	P.AII:	.ASCII	<3>\!AC\	:	
											43	41	21	03	00DBD	P.AIJ:	.ASCII	<5>\!ASTRT\	:	
											43	41	21	03	00DC3	P.AIK:	.ASCII	<3>\!AC\	:	
											43	41	21	03	00DC7	P.AIL:	.ASCII	<7>\Running\	:	
											43	41	21	03	00DCF	P.AIM:	.ASCII	<3>\!AC\	:	
											43	41	21	03	00DD3	P.AIN:	.ASCII	<11>\Maintenance\	:	
											43	41	21	03	00DDF	P.AIO:	.ASCII	<9>\Header = \	:	
75	62	69	72	74	20	64	65	74	63	65	6C	65	53	18	00DE9	P.AIP:	.ASCII	<24>\Selected tributary = !UL\	:	
											43	41	21	03	00DF8				:	
75	62	69	72	74	20	73	75	6F	69	76	65	72	50	18	00E02	P.AIQ:	.ASCII	<24>\Previous tributary = !UL\	:	
											43	41	21	03	00E11				:	
74	61	74	73	20	79	72	61	74	75	62	69	72	54	13	00E1B	P.AIR:	.ASCII	<19>\Tributary status = \	:	
											43	41	21	03	00E2A				:	
											43	41	21	03	00E2F	P.AIS:	.ASCII	<3>\!AC\	:	
											43	41	21	03	00E33	P.AIT:	.ASCII	<9>\Streaming\	:	
											43	41	21	03	00E3D	P.AIU:	.ASCII	<3>\!AC\	:	
64	6E	65	73	20	64	65	75	6E	69	74	6E	6F	43	1C	00E41	P.AIV:	.ASCII	<28>\Continued send after timeout\	:	
											43	41	21	03	00E50				:	
											43	41	21	03	00E5E	P.AIW:	.ASCII	<3>\!AC\	:	
64	6E	65	73	20	64	65	75	6E	69	74	6E	6F	43	1D	00E62	P.AIX:	.ASCII	<29>\Continued send after deselect\	:	
74	63	65	6C	65	73	65	64	20	72	65	74	66	61	20	00E71				:	
											43	41	21	03	00E80	P.AIY:	.ASCII	<3>\!AC\	:	
6E	69	6D	61	65	72	74	73	20	64	65	64	6E	45	0F	00E84	P.AIZ:	.ASCII	<15>\Ended streaming\	:	
											43	41	21	03	00E93				:	
75	62	69	72	74	20	64	65	76	69	65	63	65	52	18	00E94	P.AJA:	.ASCII	<24>\Received tributary = !UL\	:	
											43	41	21	03	00EA3				:	

3D	20	68	74	67	6E	65	6C	20	6B	63	6F	6C	42	12	00EAD	P.AJB:	.ASCII	<18>\Block length = !UL\		
											4C	55	21	20	00EBC					
20	68	74	67	6E	65	6C	20	72	65	66	66	75	42	13	00EC0	P.AJC:	.ASCII	<19>\Buffer length = !UL\		
										4C	55	21	20	3D	00ECF					
					43	41	21	20	3D	20	45	54	44	09	00ED4	P.AJD:	.ASCII	<9>\DTE = !AC\		
					20	3D	20	6E	6F	73	61	65	52	09	00EDE	P.AJE:	.ASCII	<9>\Reason = \		
											43	41	21	03	00EE8	P.AJF:	.ASCII	<3>\!AC\		
61	6D	6D	6F	63	20	72	6F	74	61	72	65	70	4F	10	00EEC	P.AJG:	.ASCII	<16>\Operator command\		
													64	6E	00EFB					
											43	41	21	03	00EFD	P.AJH:	.ASCII	<3>\!AC\		
69	74	6	7	65	70	6F	20	6C	61	6D	72	6F	4E	10	00F01	P.AJI:	.ASCII	<16>\Normal operation\		
													6E	6F	00F10					
		20	3D	20	65	74	61	74	73	20	64	6C	4F	0C	00F12	P.AJJ:	.ASCII	<12>\Old state = \		
		20	3D	20	65	74	61	74	73	20	77	65	4E	0C	00F1F	P.AJK:	.ASCII	<12>\New state = \		
											43	41	21	03	00F2C	P.AJL:	.ASCII	<3>\!AC\		
												6E	4F	02	00F30	P.AJM:	.ASCII	<2>\On\		
											43	41	21	03	00F33	P.AJN:	.ASCII	<3>\!AC\		
											66	66	4F	03	00F37	P.AJO:	.ASCII	<3>\Off\		
											43	41	21	03	00F3B	P.AJP:	.ASCII	<3>\!AC\		
										74	75	68	53	04	00F3F	P.AJQ:	.ASCII	<4>\Shut\		
65	70	79	74	20	72	65	74	65	6D	61	72	61	50	15	00F44	P.AJR:	.ASCII	<21>\Parameter type = #!UL\		
								4C	55	21	23	20	3D	20	00F53					
21	20	3D	20	63	69	74	73	6F	6E	67	61	69	44	10	00F5A	P.AJS:	.ASCII	<11>\Cause = !UL\		
													4C	55	00F66	P.AJT:	.ASCII	<16>\Diagnostic = !UL\		
6E	6F	73	61	65	72	20	65	72	75	6C	69	61	46	11	00F75					
													20	3D	20	00F77	P.AJU:	.ASCII	<17>\Failure reason = \	
											43	41	21	03	00F86					
6C	6C	6F	63	20	65	76	69	73	73	65	63	78	45	14	00F89	P.AJV:	.ASCII	<3>\!AC\		
										73	6E	6F	69	73	69	00F8D	P.AJW:	.ASCII	<20>\Excessive collisions\	
											43	41	21	03	00F9C					
20	6B	63	65	68	63	20	72	65	69	72	72	61	43	14	00FA2	P.AJX:	.ASCII	<3>\!AC\		
											64	65	6C	69	66	00FA6	P.AJY:	.ASCII	<20>\Carrier check failed\	
											43	41	21	03	00FB5					
															00FBB	P.AJZ:	.ASCII	<3>\!AC\		
															00FBF	P.AKA:	.ASCII	<0>		
											43	41	21	03	00FC0	P.AKB:	.ASCII	<3>\!AC\		
	74	69	75	63	72	69	63	20	74	72	6F	68	53	0D	00FC4	P.AKC:	.ASCII	<13>\Short circuit\		
											43	41	21	03	00FD2	P.AKD:	.ASCII	<3>\!AC\		
	74	69	75	63	72	69	63	20	6E	65	70	4F	0C	00FD6	P.AKE:	.ASCII	<12>\Open circuit\			
											43	41	21	03	00FE3	P.AKF:	.ASCII	<3>\!AC\		
67	6E	6F	6C	20	6F	6F	74	20	65	6D	61	72	46	0E	00FE7	P.AKG:	.ASCII	<14>\Frame too long\		
											43	41	21	03	00FF6	P.AKH:	.ASCII	<3>\!AC\		
65	72	75	6C	69	61	66	20	65	74	6F	6D	65	52	17	00FFA	P.AKI:	.ASCII	<23>\Remote failure to defer\		
					72	65	66	65	64	20	6F	74	20		01009					
										43	41	21	03		01012	P.AKJ:	.ASCII	<3>\!AC\		
72	65	20	6B	63	65	68	63	20	6B	63	6F	6C	42	11	01016	P.AKK:	.ASCII	<17>\Block check error\		
												72	6F	72	01025					
											43	41	21	03	01028	P.AKL:	.ASCII	<3>\!AC\		
	72	6F	72	72	65	20	67	6E	69	6D	61	72	46	0D	0102C	P.AKM:	.ASCII	<13>\Framing error\		
											43	41	21	03	0103A	P.AKN:	.ASCII	<3>\!AC\		
		6E	75	72	72	65	76	6F	20	61	74	61	44	0C	0103E	P.AKO:	.ASCII	<12>\Data overrun\		
											43	41	21	03	0104B	P.AKP:	.ASCII	<3>\!AC\		
20	72	65	66	66	75	62	20	6D	65	74	73	79	53	19	0104F	P.AKQ:	.ASCII	<25>\System buffer unavailable\		
			65	6C	62	61	6C	69	61	76	61	6E	75		0105E					
											43	41	21	03	01069	P.AKR:	.ASCII	<3>\!AC\		
6E	75	20	72	65	66	66	75	62	20	72	65	73	55	17	0106D	P.AKS:	.ASCII	<23>\User buffer unavailable\		
					65	6C	62	61	6C	69	61	76	61	76	61	0107C				

100

PC	OP	OP2	OP3	OP4	OP5	OP6	OP7	OP8	OP9	OP10	OP11	OP12	OP13	OP14	OP15	OP16	OP17	OP18	OP19	OP20	OP21	OP22	OP23	OP24	OP25	OP26	OP27	OP28	OP29	OP30	OP31	OP32	OP33	OP34	OP35	OP36	OP37	OP38	OP39	OP40	OP41	OP42	OP43	OP44	OP45	OP46	OP47	OP48	OP49	OP50	OP51	OP52	OP53	OP54	OP55	OP56	OP57	OP58	OP59	OP60	OP61	OP62	OP63	OP64	OP65	OP66	OP67	OP68	OP69	OP70	OP71	OP72	OP73	OP74	OP75	OP76	OP77	OP78	OP79	OP80	OP81	OP82	OP83	OP84	OP85	OP86	OP87	OP88	OP89	OP90	OP91	OP92	OP93	OP94	OP95	OP96	OP97	OP98	OP99	OP100	OP101	OP102	OP103	OP104	OP105	OP106	OP107	OP108	OP109	OP110	OP111	OP112	OP113	OP114	OP115	OP116	OP117	OP118	OP119	OP120	OP121	OP122	OP123	OP124	OP125	OP126	OP127	OP128	OP129	OP130	OP131	OP132	OP133	OP134	OP135	OP136	OP137	OP138	OP139	OP140	OP141	OP142	OP143	OP144	OP145	OP146	OP147	OP148	OP149	OP150	OP151	OP152	OP153	OP154	OP155	OP156	OP157	OP158	OP159	OP160	OP161	OP162	OP163	OP164	OP165	OP166	OP167	OP168	OP169	OP170	OP171	OP172	OP173	OP174	OP175	OP176	OP177	OP178	OP179	OP180	OP181	OP182	OP183	OP184	OP185	OP186	OP187	OP188	OP189	OP190	OP191	OP192	OP193	OP194	OP195	OP196	OP197	OP198	OP199	OP200	OP201	OP202	OP203	OP204	OP205	OP206	OP207	OP208	OP209	OP210	OP211	OP212	OP213	OP214	OP215	OP216	OP217	OP218	OP219	OP220	OP221	OP222	OP223	OP224	OP225	OP226	OP227	OP228	OP229	OP230	OP231	OP232	OP233	OP234	OP235	OP236	OP237	OP238	OP239	OP240	OP241	OP242	OP243	OP244	OP245	OP246	OP247	OP248	OP249	OP250	OP251	OP252	OP253	OP254	OP255	OP256	OP257	OP258	OP259	OP260	OP261	OP262	OP263	OP264	OP265	OP266	OP267	OP268	OP269	OP270	OP271	OP272	OP273	OP274	OP275	OP276	OP277	OP278	OP279	OP280	OP281	OP282	OP283	OP284	OP285	OP286	OP287	OP288	OP289	OP290	OP291	OP292	OP293	OP294	OP295	OP296	OP297	OP298	OP299	OP300	OP301	OP302	OP303	OP304	OP305	OP306	OP307	OP308	OP309	OP310	OP311	OP312	OP313	OP314	OP315	OP316	OP317	OP318	OP319	OP320	OP321	OP322	OP323	OP324	OP325	OP326	OP327	OP328	OP329	OP330	OP331	OP332	OP333	OP334	OP335	OP336	OP337	OP338	OP339	OP340	OP341	OP342	OP343	OP344	OP345	OP346	OP347	OP348	OP349	OP350	OP351	OP352	OP353	OP354	OP355	OP356	OP357	OP358	OP359	OP360	OP361	OP362	OP363	OP364	OP365	OP366	OP367	OP368	OP369	OP370	OP371	OP372	OP373	OP374	OP375	OP376	OP377	OP378	OP379	OP380	OP381	OP382	OP383	OP384	OP385	OP386	OP387	OP388	OP389	OP390	OP391	OP392	OP393	OP394	OP395	OP396	OP397	OP398	OP399	OP400	OP401	OP402	OP403	OP404	OP405	OP406	OP407	OP408	OP409	OP410	OP411	OP412	OP413	OP414	OP415	OP416	OP417	OP418	OP419
----	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

F 16
16-Sep-1984 01:31:10 VAX-11 Bliss-32 V4.0-742 Page 40
14-Sep-1984 12:28:46 DISK\$VMSMASTER:[EVL.SRC]CONSOLE.B32;1 (5)

PC	Op	OpC	OpD	OpI	OpR	OpS	OpT	OpU	OpV	OpW	OpX	OpY	OpZ	OpAA	OpAB	OpAC	OpAD	OpAE	OpAF	OpAG	OpAH	OpAI	OpAJ	OpAK	OpAL	OpAM	OpAN	OpAO	OpAP	OpAQ	OpAR	OpAS	OpAT	OpAU	OpAV	OpAW	OpAX	OpAY	OpAZ	OpBA	OpBB	OpBC	OpBD	OpBE	OpBF	OpBG	OpBH	OpBI	OpBJ	OpBK	OpBL	OpBM	OpBN	OpBO	OpBP	OpBQ	OpBR	OpBS	OpBT	OpBU	OpBV	OpBW	OpBX	OpBY	OpBZ	OpCA	OpCB	OpCC	OpCD	OpCE	OpCF	OpCG	OpCH	OpCI	OpCJ	OpCK	OpCL	OpCM	OpCN	OpCO	OpCP	OpCQ	OpCR	OpCS	OpCT	OpCU	OpCV	OpCW	OpCX	OpCY	OpCZ	OpDA	OpDB	OpDC	OpDD	OpDE	OpDF	OpDG	OpDH	OpDI	OpDJ	OpDK	OpDL	OpDM	OpDN	OpDO	OpDP	OpDQ	OpDR	OpDS	OpDT	OpDU	OpDV	OpDW	OpDX	OpDY	OpDZ	OpEA	OpEB	OpEC	OpED	OpEE	OpEF	OpEG	OpEH	OpEI	OpEJ	OpEK	OpEL	OpEM	OpEN	OpEO	OpEP	OpEQ	OpER	OpES	OpET	OpEU	OpEV	OpEW	OpEX	OpEY	OpEZ	OpFA	OpFB	OpFC	OpFD	OpFE	OpFF	OpFG	OpFH	OpFI	OpFJ	OpFK	OpFL	OpFM	OpFN	OpFO	OpFP	OpFQ	OpFR	OpFS	OpFT	OpFU	OpFV	OpFW	OpFX	OpFY	OpFZ	OpGA	OpGB	OpGC	OpGD	OpGE	OpGF	OpGG	OpGH	OpGI	OpGJ	OpGK	OpGL	OpGM	OpGN	OpGO	OpGP	OpGQ	OpGR	OpGS	OpGT	OpGU	OpGV	OpGW	OpGX	OpGY	OpGZ	OpHA	OpHB	OpHC	OpHD	OpHE	OpHF	OpHG	OpHH	OpHI	OpHJ	OpHK	OpHL	OpHM	OpHN	OpHO	OpHP	OpHQ	OpHR	OpHS	OpHT	OpHU	OpHV	OpHW	OpHX	OpHY	OpHZ	OpIA	OpIB	OpIC	OpID	OpIE	OpIF	OpIG	OpIH	OpII	OpIJ	OpIK	OpIL	OpIM	OpIN	OpIO	OpIP	OpIQ	OpIR	OpIS	OpIT	OpIU	OpIV	OpIW	OpIX	OpIY	OpIZ	OpJA	OpJB	OpJC	OpJD	OpJE	OpJF	OpJG	OpJH	OpJI	OpJJ	OpJK	OpJL	OpJM	OpJN	OpJO	OpJP	OpJQ	OpJR	OpJS	OpJT	OpJU	OpJV	OpJW	OpJX	OpJY	OpJZ	OpKA	OpKB	OpKC	OpKD	OpKE	OpKF	OpKG	OpKH	OpKI	OpKJ	OpKK	OpKL	OpKM	OpKN	OpKO	OpKP	OpKQ	OpKR	OpKS	OpKT	OpKU	OpKV	OpKW	OpKX	OpKY	OpKZ	OpLA	OpLB	OpLC	OpLD	OpLE	OpLF	OpLG	OpLH	OpLI	OpLJ	OpLK	OpLL	OpLM	OpLN	OpLO	OpLP	OpLQ	OpLR	OpLS	OpLT	OpLU	OpLV	OpLW	OpLX	OpLY	OpLZ	OpMA	OpMB	OpMC	OpMD	OpME	OpMF	OpMG	OpMH	OpMI	OpMJ	OpMK	OpML	OpMM	OpMN	OpMO	OpMP	OpMQ	OpMR	OpMS	OpMT	OpMU	OpMV	OpMW	OpMX	OpMY	OpMZ	OpNA	OpNB	OpNC	OpND	OpNE	OpNF	OpNG	OpNH	OpNI	OpNJ	OpNK	OpNL	OpNM	OpNN	OpNO	OpNP	OpNQ	OpNR	OpNS	OpNT	OpNU	OpNV	OpNW	OpNX	OpNY	OpNZ	OpOA	OpOB	OpOC	OpOD	OpOE	OpOF	OpOG	OpOH	OpOI	OpOJ	OpOK	OpOL	OpOM	OpON	OpOO	OpOP	OpOQ	OpOR	OpOS	OpOT	OpOU	OpOV	OpOW	OpOX	OpOY	OpOZ	OpPA	OpPB	OpPC	OpPD	OpPE	OpPF	OpPG	OpPH	OpPI	OpPJ	OpPK	OpPL	OpPM	OpPN	OpPO	OpPP	OpPQ	OpPR	OpPS	OpPT	OpPU	OpPV	OpPW	OpPX	OpPY	OpPZ	OpQA	OpQB	OpQC	OpQD	OpQE	OpQF	OpQG	OpQH	OpQI	OpQJ	OpQK	OpQL	OpQM	OpQN	OpQO	OpQP	OpQQ	OpQR	OpQS	OpQT	OpQU	OpQV	OpQW	OpQX	OpQY	OpQZ	OpRA	OpRB	OpRC	OpRD	OpRE	OpRF	OpRG	OpRH	OpRI	OpRJ	OpRK	OpRL	OpRM	OpRN	OpRO	OpRP	OpRQ	OpRR	OpRS	OpRT	OpRU	OpRV	OpRW	OpRX	OpRY	OpRZ	OpSA	OpSB	OpSC	OpSD	OpSE	OpSF	OpSG	OpSH	OpSI	OpSJ	OpSK	OpSL	OpSM	OpSN	OpSO	OpSP	OpSQ	OpSR	OpSS	OpST	OpSU	OpSV
----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------

					36\$-25\$.-
					37\$-25\$.-
					38\$-25\$.-
					39\$-25\$.-
					41\$-25\$.-
					43\$-25\$.-
					45\$-25\$.-
					47\$-25\$.-
					49\$-25\$.-
					51\$-25\$.-
					53\$-25\$.-
					55\$-25\$.-
					57\$-25\$.-
					59\$-25\$.-
					61\$-25\$.-
					63\$-25\$.-
					65\$-25\$.-
					67\$-25\$.-
					69\$-25\$.-
0000'	CF	9F	00122	26\$:	PUSHAB P.AAZ
0000'	CF	9F	00126		PUSHAB P.AAY
	008C	31	0012A		BRW 40\$
0000'	CF	9F	0012D	27\$:	PUSHAB P.ABB
0000'	CF	9F	00131		PUSHAB P.ABA
	008C	31	00135		BRW 42\$
0000'	CF	9F	00138	28\$:	PUSHAB P.ABD
0000'	CF	9F	0013C		PUSHAB P.ABC
	008C	31	00140		BRW 44\$
0000'	CF	9F	00143	29\$:	PUSHAB P.ABF
0000'	CF	9F	00147		PUSHAB P.ABE
	008B	31	0014B		BRW 46\$
0000'	CF	9F	0014E	30\$:	PUSHAB P.ABH
0000'	CF	9F	00152		PUSHAB P.ABG
	008A	31	00156		BRW 48\$
0000'	CF	9F	00159	31\$:	PUSHAB P.ABJ
0000'	CF	9F	0015D		PUSHAB P.ABI
	0089	31	00161		BRW 50\$
0000'	CF	9F	00164	32\$:	PUSHAB P.ABL
0000'	CF	9F	00168		PUSHAB P.ABK
	0088	31	0016C		BRW 52\$
0000'	CF	9F	0016F	33\$:	PUSHAB P.ABN
0000'	CF	9F	00173		PUSHAB P.ABM
	0087	31	00177		BRW 54\$
0000'	CF	9F	0017A	34\$:	PUSHAB P.ABP
0000'	CF	9F	0017E		PUSHAB P.ABO
	0086	31	00182		BRW 56\$
0000'	CF	9F	00185	35\$:	PUSHAB P.ABR
0000'	CF	9F	00189		PUSHAB P.ABQ
	0085	31	0018D		BRW 58\$
0000'	CF	9F	00190	36\$:	PUSHAB P.ABT
0000'	CF	9F	00194		PUSHAB P.ABS
	0084	31	00198		BRW 60\$
0000'	CF	9F	0019B	37\$:	PUSHAB P.ABV
0000'	CF	9F	0019F		PUSHAB P.ABU
	0083	31	001A3		BRW 62\$
0000'	CF	9F	001A6	38\$:	PUSHAB P.ABX
0000'	CF	9F	001AA		PUSHAB P.ABW

		0082	31	001AE	BRW	64\$	
0000'	CF	9F	001B1	39\$:	PUSHAB	P.ABZ	
0000'	CF	9F	001B5		PUSHAB	P.ABY	
		0081	31	001B9	BRW	66\$	
0000'	CF	9F	001BC	40\$:	PUSHAB	P.ACB	
0000'	CF	9F	001C0	41\$:	PUSHAB	P.ACA	
		0080	31	001C4	BRW	68\$	
0000'	CF	9F	001C7	42\$:	PUSHAB	P.ACD	
0000'	CF	9F	001CB	43\$:	PUSHAB	P.ACC	
		76	11	001CF	BRB	68\$	
0000'	CF	9F	001D1	44\$:	PUSHAB	P.ACF	
0000'	CF	9F	001D5	45\$:	PUSHAB	P.ACE	
		76	11	001D9	BRB	70\$	
0000'	CF	9F	001DB	46\$:	PUSHAB	P.ACH	
0000'	CF	9F	001DF	47\$:	PUSHAB	P.ACG	
		6C	11	001E3	BRB	70\$	
0000'	CF	9F	001E5	48\$:	PUSHAB	P.ACJ	
0000'	CF	9F	001E9	49\$:	PUSHAB	P.ACI	
		62	11	001ED	BRB	70\$	
0000'	CF	9F	001EF	50\$:	PUSHAB	P.ACL	
0000'	CF	9F	001F3	51\$:	PUSHAB	P.ACK	
		58	11	001F7	BRB	70\$	
0000'	CF	9F	001F9	52\$:	PUSHAB	P.ACN	
0000'	CF	9F	001FD	53\$:	PUSHAB	P.ACM	
		4E	11	00201	BRB	70\$	
0000'	CF	9F	00203	54\$:	PUSHAB	P.ACP	
0000'	CF	9F	00207	55\$:	PUSHAB	P.ACO	
		44	11	0020B	BRB	70\$	
0000'	CF	9F	0020D	56\$:	PUSHAB	P.ACR	
0000'	CF	9F	00211	57\$:	PUSHAB	P.ACQ	
		3A	11	00215	BRB	70\$	
0000'	CF	9F	00217	58\$:	PUSHAB	P.ACT	
0000'	CF	9F	0021B	59\$:	PUSHAB	P.ACS	
		30	11	0021F	BRB	70\$	
0000'	CF	9F	00221	60\$:	PUSHAB	P.ACV	
0000'	CF	9F	00225	61\$:	PUSHAB	P.ACU	
		26	11	00229	BRB	70\$	
0000'	CF	9F	0022B	62\$:	PUSHAB	P.ACX	
0000'	CF	9F	0022F	63\$:	PUSHAB	P.ACW	
		1C	11	00233	BRB	70\$	
0000'	CF	9F	00235	64\$:	PUSHAB	P.ACZ	
0000'	CF	9F	00239	65\$:	PUSHAB	P.ACY	
		12	11	0023D	BRB	70\$	
0000'	CF	9F	0023F	66\$:	PUSHAB	P.ADB	
0000'	CF	9F	00243	67\$:	PUSHAB	P.ADA	
		08	11	00247	BRB	70\$	
0000'	CF	9F	00249	68\$:	PUSHAB	P.ADD	
0000'	CF	9F	0024D	69\$:	PUSHAB	P.ADC	
		53	DD	00251	PUSHL	R3	
FD6F	CF	02	03	FB	CALLS	#3, APPEND_STRINGS	
			5A	D1	CMPL	FIELDS, #2	1159
			03	18	BGEQ	72\$	
			D110	31	BRW	100\$	
F3	8F		57	91	CMPB	BYTE2, #-13	1164
			58	12	BNEQ	80\$	
0000'	CF	9F	00266		PUSHAB	P.ADE	
		53	DD	0026A	PUSHL	R3	

002A	05 0020	FD56 CF 00 0016 003E	06 02 FB 0026C A2 8F 00271 000C 00276 73\$: 0034 0027E	CALLS #2, APPEND STRINGS CASEB 6(PTR) #0, #5 .WORD 74\$-73\$,- 75\$-73\$,- 76\$-73\$,- 77\$-73\$,- 78\$-73\$,- 79\$-73\$	1171
			0000' CF 9F 00282 74\$: 0000' CF 9F 00286 7C 11 0028A 0000' CF 9F 0028C 75\$: 0000' CF 9F 00290 7C 11 00294 0000' CF 9F 00296 76\$: 0000' CF 9F 0029A 7C 11 0029E 0000' CF 9F 002A0 77\$: 0000' CF 9F 002A4 7C 11 002A8 0000' CF 9F 002AA 78\$: 0000' CF 9F 002AE 72 11 002B2 0000' CF 9F 002B4 79\$: 0000' CF 9F 002B8 68 11 002BC 57 91 002BE 80\$: 0C 13 002C2 57 91 002C4 65 19 002C8 57 91 002CA 5F 14 002CE 0000' CF 9F 002D0 81\$: 53 DD 002D4 02 FB 002D6 A2 8F 002DB 000C 002E0 82\$: 0034 002E8	PUSHAB P.ADG PUSHAB P.ADF BRB 86\$ PUSHAB P.ADI PUSHAB P.ADH BRB 88\$ PUSHAB P.ADK PUSHAB P.ADJ BRB 90\$ PUSHAB P.ADM PUSHAB P.ADL BRB 92\$ PUSHAB P.ADO PUSHAB P.ADN BRB 92\$ PUSHAB P.ADQ PUSHAB P.ADP BRB 92\$ CMPB BYTE2, #-11 BEQL 81\$ CMPB BYTE2, #-9 BLSS 93\$ CMPB BYTE2, #-8 BGTR 93\$ PUSHAB P.ADR PUSHL R3 CALLS #2, APPEND STRINGS CASEB 6(PTR) #0, #5 .WORD 83\$-82\$,- 84\$-82\$,- 85\$-82\$,- 87\$-82\$,- 89\$-82\$,- 91\$-82\$	
		F5 8F F7 8F F8 8F			1172
			0000' CF 9F 002EC 83\$: 0000' CF 9F 002F0 30 11 002F4 0000' CF 9F 002F6 84\$: 0000' CF 9F 002FA 26 11 002FE 0000' CF 9F 00300 85\$: 0000' CF 9F 00304 1C 11 00308 86\$: 0000' CF 9F 0030A 87\$: 0000' CF 9F 0030E 12 11 00312 88\$: 0000' CF 9F 00314 89\$: 0000' CF 9F 00318 08 11 0031C 90\$:	PUSHAB P.ADT PUSHAB P.ADS BRB 92\$ PUSHAB P.ADV PUSHAB P.ADU BRB 92\$ PUSHAB P.ADX PUSHAB P.ADW BRB 92\$ PUSHAB P.ADZ PUSHAB P.ADY BRB 92\$ PUSHAB P.AEB PUSHAB P.AEA BRB 92\$	1173
002A	05 0020	FCEC CF 00 0016 003E	06 02 FB 0026C A2 8F 00271 000C 00276 82\$: 0034 0027E		1180

		0000'	CF	9F	0031E	91\$:	PUSHAB	P.AED	
		0000'	CF	9F	00322		PUSHAB	P.AEC	
			53	DD	00326	92\$:	PUSHL	R3	
	FC9A	CF	03	FB	00328		CALLS	#3, APPEND_STRINGS	
			41	11	0032D		BRB	100\$	
	E3	8F	57	91	0032F	93\$:	CMPB	BYTE2, #-29	1181
			0C	13	00333		BEQL	94\$	
	F0	8F	57	91	00335		CMPB	BYTE2, #-16	
			06	13	00339		BEQL	94\$	
	FA	8F	57	91	0033B		CMPB	BYTE2, #-6	
			2F	12	0033F		BNEQ	100\$	
		0000'	CF	9F	00341	94\$:	PUSHAB	P.AEE	1182
			53	DD	00345		PUSHL	R3	
	FC7B	CF	02	FB	00347		CALLS	#2, APPEND_STRINGS	
02	0078	8F	06	A2	0034C		CASEW	6(PTR), #120, #2	1183
0012		000C	0006	AF	00353	95\$:	.WORD	96\$-95\$,- 97\$-95\$,- 98\$-95\$	
		0000'	CF	9F	00359	96\$:	PUSHAB	P.AEF	118
			0A	11	0035D		BRB	99\$	
		0000'	CF	9F	0035F	97\$:	PUSHAB	P.AEG	1188
			04	11	00363		BRB	99\$	
		0000'	CF	9F	00365	98\$:	PUSHAB	P.AEH	1189
			53	DD	00369	99\$:	PUSHL	R3	
	FC57	CF	02	FB	0036B		CALLS	#2, APPEND_STRINGS	
		03	5A	D1	00370	100\$:	CMPL	FIELDS, #3	1193
			03	18	00373		BGEQ	102\$	
		08F9	31	00375	101\$:	BRW	279\$		
		09	A2	95	00378	102\$:	TSTB	9(PTR)	1194
			F8	13	0037B		BEQL	101\$	
		09	A2	9F	0037D		PUSHAB	9(PTR)	1196
		0000'	CF	9F	00380		PUSHAB	P.AEI	
			70	11	00384		BRB	114\$	
		02	54	D1	00386	103\$:	CMPL	PARAM_TYPE, #2	1198
			28	12	00389		BNEQ	107\$	
		0000'	CF	9F	0038B		PUSHAB	P.AEJ	
			53	DD	0038F		PUSHL	R3	
	FC31	CF	02	FB	00391		CALLS	#2, APPEND_STRINGS	
01		00	58	8F	00396		CASEB	BYTE1, #0, #1	1199
		000E	0004		0039A	104\$:	.WORD	105\$-104\$,- 106\$-104\$	
		0000'	CF	9F	0039E	105\$:	PUSHAB	P.AEL	
		0000'	CF	9F	003A2		PUSHAB	P.AEK	
			7E	11	003A6		BRB	117\$	
		0000'	CF	9F	003AB	106\$:	PUSHAB	P.AEN	
		0000'	CF	9F	003AC		PUSHAB	P.AEM	
			0080	31	003B0		BRW	119\$	
		03	54	D1	003B3	107\$:	CMPL	PARAM_TYPE, #3	1200
			40	12	003B6		BNEQ	115\$	
		04	58	8F	003B8		CASEB	BYTE1, #0, #4	1206
0028	001E	0014	000A		003BC	108\$:	.WORD	109\$-108\$,- 110\$-108\$,- 111\$-108\$,- 112\$-108\$,- 113\$-108\$	
			0032		003C4				
		0000'	CF	9F	003C6	109\$:	PUSHAB	P.AEP	
		0000'	CF	9F	003CA		PUSHAB	P.AEO	

		70	11	003CE	BRB	121\$	
	0000'	CF	9F	003D0	110\$:	PUSHAB	P.AER
	0000'	CF	9F	003D4		PUSHAB	P.AEQ
		7F	11	003D8		BRB	125\$
	0000'	CF	9F	003DA	111\$:	PUSHAB	P.AET
	0000'	CF	9F	003DE		PUSHAB	P.AES
		7F	11	003E2		BRB	127\$
	0000'	CF	9F	003E4	112\$:	PUSHAB	P.AEV
	0000'	CF	9F	003E8		PUSHAB	P.AEU
		7F	11	003EC		BRB	129\$
	0000'	CF	9F	003EE	113\$:	PUSHAB	P.AEX
	0000'	CF	9F	003F2		PUSHAB	P.AEW
		75	11	003F6	114\$:	BRB	129\$
	05	54	D1	003F8	115\$:	CMPL	PARAM_TYPE, #5
		2B	12	003FB		BNEQ	118\$
	7E	69	3C	003FD		MOVZWL	(R9), -(SP)
0000V	CF	01	FB	00400		CALLS	#1, FORMAT_NODEADR
		50	DD	00405		PUSHL	R0
	0000'	CF	9F	00407		PUSHAB	P.AEY
		53	DD	0040B		PUSHL	R3
FBB5	CF	03	FB	0040D		CALLS	#3, APPEND_STRINGS
	01	5A	D1	00412		CMPL	FIELDS, #1
		03	14	00415		BGTR	116\$
		0857	31	00417		BRW	279\$
	07	A2	95	0041A	116\$:	TSTB	7(PTR)
		7E	13	0041D		BEQL	133\$
	07	A2	9F	0041F		PUSHAB	7(PTR)
	0000'	CF	9F	00422		PUSHAB	P.AEZ
		45	11	00426	117\$:	BRB	129\$
	06	54	D1	00428	118\$:	CMPL	PARAM_TYPE, #6
		08	12	0042B		BNEQ	120\$
		5B	DD	0042D		PUSHL	R11
	0000'	CF	9F	0042F		PUSHAB	P.AFA
		38	11	00433	119\$:	BRB	129\$
	07	54	D1	00435	120\$:	CMPL	PARAM_TYPE, #7
		08	12	00438		BNEQ	122\$
		5B	DD	0043A		PUSHL	R11
	0000'	CF	9F	0043C		PUSHAB	P.AFB
		2B	11	00440	121\$:	BRB	129\$
	08	54	D1	00442	122\$:	CMPL	PARAM_TYPE, #8
		29	12	00445		BNEQ	130\$
02	00	58	8F	00447		CASEB	BYTE1, #0, #2
001A	0010	0006		0044B	123\$:	.WORD	124\$-123\$,-
							126\$-123\$,-
							128\$-123\$
	0000'	CF	9F	00451	124\$:	PUSHAB	P.AFD
	0000'	CF	9F	00455		PUSHAB	P.AFC
		12	11	00459	125\$:	BRB	129\$
	0000'	CF	9F	0045B	126\$:	PUSHAB	P.AFF
	0000'	CF	9F	0045F		PUSHAB	P.AFE
		08	11	00463	127\$:	BRB	129\$
	0000'	CF	9F	00465	128\$:	PUSHAB	P.AFH
	0000'	CF	9F	00469		PUSHAB	P.AFG
		01DB	31	0046D	129\$:	BRW	161\$
	09	54	D1	00470	130\$:	CMPL	PARAM_TYPE, #9
		03	13	00473		BEQL	131\$
		07ED	31	00475		BRW	278\$

	7E		0000'	69	9A	00478	131\$:	MOVZBL	(R9) -(SP)		1222
				CF	9F	00478		PUSHAB	P.AFI		
				53	DD	0047F		PUSHL	R3		
FB41	CF			03	FB	00481		CALLS	#3, APPEND_STRINGS		
	57			01	D0	00486		MOVL	#1, P		1223
	7E		04	A742	9A	00489	132\$:	MOVZBL	4(P)[PTR], -(SP)		1224
			0000'	CF	9F	0048E		PUSHAB	P.AFJ		
				53	DD	00492		PUSHL	R3		
FB2E	CF			03	FB	00494		CALLS	#3, APPEND_STRINGS		
EC	57			05	F3	00499		AOBLEQ	#5, P, 132\$		
			07	D1	31	0049D	133\$:	BRW	279\$		1118
	02			55	D1	004A0	134\$:	CMPL	R5, #2		1232
				03	13	004A3		BEQL	135\$		
			01	50	31	004A5		BRW	157\$		
	57			01	D0	004A8	135\$:	MOVL	#1, R7		1233
				54	D5	004AB		TSTL	PARAM_TYPE		1236
				23	12	004AD		BNEQ	140\$		
01				57	D4	004AF		CLRL	R7		
	00			58	8F	004B1		CASEB	BYTE1, #0, #1		1237
	000E		0004			004B5	136\$:	.WORD	137\$-136\$,- 138\$-136\$		
			0000'	CF	9F	004B9	137\$:	PUSHAB	P.AFL		
			0000'	CF	9F	004BD		PUSHAB	P.AFK		
				08	11	004C1		BRB	139\$		
			0000'	CF	9F	004C3	138\$:	PUSHAB	P.AFN		
			0000'	CF	9F	004C7		PUSHAB	P.AFM		
				53	DD	004CB	139\$:	PUSHL	R3		
FAF5	CF			03	FB	004CD		CALLS	#3, APPEND_STRINGS		
	01			54	D1	004D2	140\$:	CMPL	PARAM_TYPE, #1		1238
				0D	12	004D5		BNEQ	141\$		
				57	D4	004D7		CLRL	R7		
			0000'	CF	9F	004D9		PUSHAB	P.AFO		
				53	DD	004DD		PUSHL	R3		
FAE3	CF			02	FB	004DF		CALLS	#2, APPEND_STRINGS		
	02			54	D1	004E4	141\$:	CMPL	PARAM_TYPE, #2		1239
				0D	12	004E7		BNEQ	142\$		
				57	D4	004E9		CLRL	R7		
			0000'	CF	9F	004EB		PUSHAB	P.AFP		
				53	DD	004EF		PUSHL	R3		
FAD1	CF			02	FB	004F1		CALLS	#2, APPEND_STRINGS		
				54	D5	004F6	142\$:	TSTL	PARAM_TYPE		1240
				40	13	004F8		BEQL	149\$		
	02			54	D1	004FA		CMPL	PARAM_TYPE, #2		
				3B	1A	004FD		BGTRU	149\$		
				57	D4	004FF		CLRL	R7		
				58	8F	00501		CASEB	BYTE1, #0, #3		1241
0026		03				00505	143\$:	.WORD	144\$-143\$,- 145\$-143\$,- 146\$-143\$,- 147\$-143\$		
	001C		0012								
			0000'	CF	9F	0050D	144\$:	PUSHAB	P.AFR		
			0000'	CF	9F	00511		PUSHAB	P.AFQ		
				1C	11	00515		BRB	148\$		
			0000'	CF	9F	00517	145\$:	PUSHAB	P.AFT		
			0000'	CF	9F	0051B		PUSHAB	P.AFS		
				12	11	0051F		BRB	148\$		
			0000'	CF	9F	00521	146\$:	PUSHAB	P.AFV		

		0000'	CF	9F	00525	PUSHAB	P.AFU		
			08	11	00529	BRB	148\$		
		0000'	CF	9F	0052B	147\$:	PUSHAB	P.AFX	
		0000'	CF	9F	0052F		PUSHAB	P.AFW	
FA8D	CF		53	DD	00533	148\$:	PUSHL	R3	
	03		03	FB	00535		CALLS	#3, APPEND_STRINGS	
			54	D1	0053A	149\$:	CMPL	PARAM_TYPE, #3	1242
			2F	12	0053D		BNEQ	150\$	
			57	D4	0053F		CLRL	R7	
	7E		69	3C	00541		MOVZWL	(R9), -(SP)	1243
0000V	CF		01	FB	00544		CALLS	#1, FORMAT_NODEADR	
			50	DD	00549		PUSHL	R0	
		0000'	CF	9F	0054B		PUSHAB	P.AFY	
			53	DD	0054F		PUSHL	R3	
FA71	CF		03	FB	00551		CALLS	#3, APPEND_STRINGS	
	01		5A	D1	00556		CMPL	FIELDS, #1	1244
			13	15	00559		BLEQ	150\$	
		07	A2	95	0055B		TSTB	7(PTR)	1245
			0E	13	0055E		BEQL	150\$	
		07	A2	9F	00560		PUSHAB	7(PTR)	1246
		0000'	CF	9F	00563		PUSHAB	P.AFZ	
			53	DD	00567		PUSHL	R3	
FA59	CF		03	FB	00569		CALLS	#3, APPEND_STRINGS	
	04		54	D1	0056E	150\$:	CMPL	PARAM_TYPE, #4	1247
			0D	12	00571		BNEQ	151\$	
			57	D4	00573		CLRL	R7	
		0000'	CF	9F	00575		PUSHAB	P.AGA	
			53	DD	00579		PUSHL	R3	
FA47	CF		02	FB	0057B		CALLS	#2, APPEND_STRINGS	
	05		54	D1	00580	151\$:	CMPL	PARAM_TYPE, #5	1248
			0D	12	00583		BNEQ	152\$	
			57	D4	00585		CLRL	R7	
		0000'	CF	9F	00587		PUSHAB	P.AGB	
			53	DD	0058B		PUSHL	R3	
FA35	CF		02	FB	0058D		CALLS	#2, APPEND_STRINGS	
	04		54	D1	00592	152\$:	CMPL	PARAM_TYPE, #4	1249
			28	1F	00595		BLSSU	153\$	
	05		54	D1	00597		CMPL	PARAM_TYPE, #5	
			23	1A	0059A		BGTRU	153\$	
			57	D4	0059C		CLRL	R7	
	7E		69	9A	0059E		MOVZBL	(R9), -(SP)	1250
		0000'	CF	9F	005A1		PUSHAB	P.AGC	
			53	DD	005A5		PUSHL	R3	
FA1B	CF		03	FB	005A7		CALLS	#3, APPEND_STRINGS	
	03		5A	D1	005AC		CMPL	FIELDS, #3	1251
			0E	15	005AF		BLEQ	153\$	
		0A	A2	9F	005B1		PUSHAB	10(PTR)	1252
		0000'	CF	9F	005B4		PUSHAB	P.AGD	
			53	DD	005B8		PUSHL	R3	
FA08	CF		03	FB	005BA		CALLS	#3, APPEND_STRINGS	
	06		54	D1	005BF	153\$:	CMPL	PARAM_TYPE, #6	1253
			0F	12	005C2		BNEQ	154\$	
			57	D4	005C4		CLRL	R7	
			5B	DD	005C6		PUSHL	R11	
		0000'	CF	9F	005C8		PUSHAB	P.AGE	
			53	DD	005CC		PUSHL	R3	
F9F4	CF		03	FB	005CE		CALLS	#3, APPEND_STRINGS	

07		54	D1	005D3	154\$:	CMPL	PARAM_TYPE, #7	1254
		0D	12	005D6		BNEQ	155\$	
		57	D4	005D8		CLRL	R7	
	0000'	CF	9F	005DA		PUSHAB	P, AGF	
F9E2	CF	53	DD	005DE		PUSHL	R3	
08		02	FB	005E0		CALLS	#2, APPEND_STRINGS	
		54	D1	005E5	155\$:	CMPL	PARAM_TYPE, #8	1255
		03	13	005E8		BEQL	156\$	
	0675	31	005EA		BRW	277\$		
		57	D4	005ED	156\$:	CLRL	R7	
		5B	DD	005EF		PUSHL	R11	
	0000'	CF	9F	005F1		PUSHAB	P, AGG	
	0663	31	005F5		BRW	276\$		
03		55	D1	005F8	157\$:	CMPL	R5, #3	1259
		58	12	005FB		BNEQ	163\$	
		54	D5	005FD		TSTL	PARAM_TYPE	1263
	0000'	0E	12	005FF		BNEQ	158\$	
		CF	9F	00601		PUSHAB	P, AGH	1264
		53	DD	00605		PUSHL	R3	
F98B	CF	02	FB	00607		CALLS	#2, APPEND_STRINGS	
		05C9	31	0060C		BRW	269\$	1265
01		54	D1	0060F	158\$:	CMPL	PARAM_TYPE, #1	1266
		09	12	00612		BNEQ	159\$	
7E		58	9A	00614		MOVZBL	BYTE1, -(SP)	
	0000'	CF	9F	00617		PUSHAB	P, AGI	
		2E	11	0061B		BRB	161\$	
02		54	D1	0061D	159\$:	CMPL	PARAM_TYPE, #2	1267
		03	13	00620		BEQL	160\$	
	0640	31	00622		BRW	278\$		
7E		69	3C	00625	160\$:	MOVZWL	(R9), -(SP)	1268
0000V	CF	01	FB	00628		CALLS	#1, FORMAT_NODEADR	
		50	DD	0062D		PUSHL	R0	
	0000'	CF	9F	0062F		PUSHAB	P, AGJ	
		53	DD	00633		PUSHL	R3	
F98D	CF	03	FB	00635		CALLS	#3, APPEND_STRINGS	
01		5A	D1	0063A		CMPL	FIELDS, #1	1269
		13	15	0063D		BLEQ	162\$	
	07	A2	95	0063F		TSTB	7(PTR)	1270
		0E	13	00642		BEQL	162\$	
	07	A2	9F	00644		PUSHAB	7(PTR)	1271
	0000'	CF	9F	00647		PUSHAB	P, AGK	
		53	DD	0064B	161\$:	PUSHL	R3	
F975	CF	03	FB	0064D		CALLS	#3, APPEND_STRINGS	
		061C	31	00652	162\$:	BRW	279\$	1269
04		55	D1	00655	163\$:	CMPL	R5, #4	1275
		03	13	00658		BEQL	164\$	
		D1D5	31	0065A		BRW	200\$	
57		01	D0	0065D	164\$:	MOVVL	#1, R7	1276
		54	D5	00660		TSTL	PARAM_TYPE	1279
		1A	12	00662		BNEQ	165\$	
		57	D4	00664		CLRL	R7	
	0000'	CF	9F	00666		PUSHAB	P, AGL	1280
		53	DD	0066A		PUSHL	R3	
F956	CF	02	FB	0066C		CALLS	#2, APPEND_STRINGS	
		0C	AC	DD	00671	PUSHL	MAXLEN	1281
	04	AE	DD	00674		PUSHL	4(SP)	
		53	DD	00677		PUSHL	R3	

0000V	CF		03	FB	00679	CALLS	#3, UNKNOWN_PARAM_DATA	
	01		54	D1	0067E	CMPL	PARAM_TYPE, #1	1282
			1A	12	00681	BNEQ	166\$	
		0000'	57	D4	00683	CLRL	R7	
			CF	9F	00685	PUSHAB	P, AGH	1283
			53	DD	00689	PUSHL	R3	
F937	CF		02	FB	0068B	CALLS	#2, APPEND_STRINGS	
		0C	AC	DD	00690	PUSHL	MAXLEN	1284
		04	AE	DD	00693	PUSHL	4(SP)	
			53	DD	00696	PUSHL	R3	
0000V	CF		03	FB	00698	CALLS	#3, UNKNOWN_PARAM_DATA	
	02		54	D1	0069D	CMPL	PARAM_TYPE, #2	1285
			10	12	006A0	BNEQ	167\$	
			57	D4	006A2	CLRL	R7	
	7E		6B	3C	006A4	MOVZWL	(R11), -(SP)	
		0000'	CF	9F	006A7	PUSHAB	P, AGN	
			53	DD	006AB	PUSHL	R3	
F915	CF		03	FB	006AD	CALLS	#3, APPEND_STRINGS	
	03		54	D1	006B2	CMPL	PARAM_TYPE, #3	1286
			0D	12	006B5	BNEQ	168\$	
		0000'	57	D4	006B7	CLRL	R7	
			CF	9F	006B9	PUSHAB	P, AGO	
			53	DD	006BD	PUSHL	R3	
F903	CF		02	FB	006BF	CALLS	#2, APPEND_STRINGS	
	08		54	D1	006C4	CMPL	PARAM_TYPE, #8	1287
			0D	12	006C7	BNEQ	169\$	
		0000'	57	D4	006C9	CLRL	R7	
			CF	9F	006CB	PUSHAB	P, AGP	
			53	DD	006CF	PUSHL	R3	
F8F1	CF		02	FB	006D1	CALLS	#2, APPEND_STRINGS	
	04		54	D1	006D6	CMPL	PARAM_TYPE, #4	1288
			0D	12	006D9	BNEQ	170\$	
		0000'	57	D4	006DB	CLRL	R7	
			CF	9F	006DD	PUSHAB	P, AGQ	
			53	DD	006E1	PUSHL	R3	
F8DF	CF		02	FB	006E3	CALLS	#2, APPEND_STRINGS	
	03		54	D1	006E8	CMPL	PARAM_TYPE, #3	1289
			05	1F	006EB	BLSSU	171\$	
	04		54	D1	006ED	CMPL	PARAM_TYPE, #4	
			05	1B	006F0	BLEQU	172\$	
	08		54	D1	006F2	CMPL	PARAM_TYPE, #8	
			2F	12	006F5	BNEQ	173\$	
			57	D4	006F7	CLRL	R7	
	7E		69	3C	006F9	MOVZWL	(R9), -(SP)	1290
0000V	CF		01	FB	006FC	CALLS	#1, FORMAT_NODEADR	
		0000'	50	DD	00701	PUSHL	R0	
			CF	9F	00703	PUSHAB	P, AGR	
			53	DD	00707	PUSHL	R3	
F889	CF		03	FB	00709	CALLS	#3, APPEND_STRINGS	
	01		5A	D1	0070E	CMPL	FIELDS, #1	1291
			13	15	00711	BLEQ	173\$	
		07	73	95	00713	TSTB	7(PTR)	1292
			0E	13	00716	BEQL	173\$	
		0000'	A2	CF	00718	PUSHAB	7(PTR)	1293
			CF	9F	0071B	PUSHAB	P, AGS	
			53	DD	0071F	PUSHL	R3	
F8A1	CF		03	FB	00721	CALLS	#3, APPEND_STRINGS	

003D	0033	0029	001E	00726	173\$:	CMPL	PARAM_TYPE, #5	1294
0065	005B	0051	0047	00729		BEQL	174\$	
008D	0083	0079	006F	0072B		BRW	193\$	
	00AB	00A1	0097	0072E	174\$:	CLRL	R7	
				00730		CASEB	BYTE1, #0, #14	1310
				00734	175\$:	.WORD	176\$-175\$,-	
				0073C			177\$-175\$,-	
				00744			178\$-175\$,-	
				0074C			179\$-175\$,-	
							180\$-175\$,-	
							181\$-175\$,-	
							182\$-175\$,-	
							183\$-175\$,-	
							184\$-175\$,-	
							185\$-175\$,-	
							186\$-175\$,-	
							187\$-175\$,-	
							188\$-175\$,-	
							189\$-175\$,-	
							191\$-175\$	
				00752	176\$:	PUSHAB	P.AGU	
				00756		PUSHAB	P.AGT	
				0075A		BRW	190\$	
				0075D	177\$:	PUSHAB	P.AGW	
				00761		PUSHAB	P.AGV	
				00765		BRB	190\$	
				00767	178\$:	PUSHAB	P.AGY	
				0076B		PUSHAB	P.AGX	
				0076F		BRB	192\$	
				00771	179\$:	PUSHAB	P.AHA	
				00775		PUSHAB	P.AGZ	
				00779		BRB	192\$	
				0077B	180\$:	PUSHAB	P.AHC	
				0077F		PUSHAB	P.AHB	
				00783		BRB	192\$	
				00785	181\$:	PUSHAB	P.AHE	
				00789		PUSHAB	P.AHD	
				0078D		BRB	192\$	
				0078F	182\$:	PUSHAB	P.AHG	
				00793		PUSHAB	P.AHF	
				00797		BRB	192\$	
				00799	183\$:	PUSHAB	P.AHI	
				0079D		PUSHAB	P.AHH	
				007A1		BRB	192\$	
				007A3	184\$:	PUSHAB	P.AHK	
				007A7		PUSHAB	P.AHJ	
				007AB		BRB	192\$	
				007AD	185\$:	PUSHAB	P.AHM	
				007B1		PUSHAB	P.AHL	
				007B5		BRB	192\$	
				007B7	186\$:	PUSHAB	P.AHO	
				007BB		PUSHAB	P.AHN	
				007BF		BRB	192\$	
				007C1	187\$:	PUSHAB	P.AHQ	
				007C5		PUSHAB	P.AHP	
				007C9		BRB	192\$	
				007CB	188\$:	PUSHAB	P.AHS	

		0000'	CF	9F	007CF	PUSHAB	P.AHR		
			12	11	007D3	BRB	192\$		
		0000'	CF	9F	007D5	189\$:	PUSHAB	P.AHU	
		0000'	CF	9F	007D9		PUSHAB	P.AHT	
			08	11	007DD	190\$:	BRB	192\$	
		0000'	CF	9F	007DF	191\$:	PUSHAB	P.AHW	
		0000'	CF	9F	007E3		PUSHAB	P.AHV	
			53	DD	007E7	192\$:	PUSHL	R3	
F7D9	CF		03	FB	007E9		CALLS	#3, APPEND_STRINGS	
	06		54	D1	007EE	193\$:	CMPL	PARAM_TYPE, #6	1311
			18	12	007F1		BNEQ	194\$	
			57	D4	007F3		CLRL	R7	
	7E	08	A2	9A	007F5		MOVZBL	8(PTR), -(SP)	1314
	7E	06	A2	9A	007F9		MOVZBL	6(PTR), -(SP)	
	7E		69	9A	007FD		MOVZBL	(R9), -(SP)	
		0000'	CF	9F	00800		PUSHAB	P.AHX	
			53	DD	00804		PUSHL	R3	
F7BC	CF		05	FB	00806		CALLS	#5, APPEND_STRINGS	
	07		54	D1	0080B	194\$:	CMPL	PARAM_TYPE, #7	1315
			03	13	0080E		BEQL	195\$	
		044F	31	00810		BRW	277\$		
			57	D4	00813	195\$:	CLRL	R7	
01	00		58	8F	00815		CASEB	BYTE1, #0, #1	1316
	000E	0004			00819	196\$:	.WORD	197\$-196\$,- 198\$-196\$	
		0000'	CF	9F	0081D	197\$:	PUSHAB	P.AHZ	
		0000'	CF	9F	00821		PUSHAB	P.AHY	
			08	11	00825		BRB	199\$	
		0000'	CF	9F	00827	198\$:	PUSHAB	P.AIB	
		0000'	CF	9F	0082B		PUSHAB	P.AIA	
		0429	31	0082F	199\$:	BRW	276\$		
	05		55	D1	00832	200\$:	CMPL	R5, #5	1320
			03	13	00835		BEQL	201\$	
		0338	31	00837		BRW	262\$		
	57		01	D0	0083A	201\$:	MOVL	#1, R7	1321
			54	D5	0083D		TSTL	PARAM_TYPE	1324
			0D	12	0083F		BNEQ	202\$	
			57	D4	00841		CLRL	R7	
		0000'	CF	9F	00843		PUSHAB	P.AIC	
			53	DD	00847		PUSHL	R3	
F779	CF		02	FB	00849		CALLS	#2, APPEND_STRINGS	
	01		54	D1	0084E	202\$:	CMPL	PARAM_TYPE, #1	1325
			0D	12	00851		BNEQ	203\$	
			57	D4	00853		CLRL	R7	
		0000'	CF	9F	00855		PUSHAB	P.AID	
			53	DD	00859		PUSHL	R3	
F767	CF		02	FB	0085B		CALLS	#2, APPEND_STRINGS	
	01		54	D1	00860	203\$:	CMPL	PARAM_TYPE, #1	1326
			47	1A	00863		BGTRU	211\$	
			57	D4	00865		CLRL	R7	
			58	8F	00867		CASEB	BYTE1, #0, #4	1328
0028	04		000A		0086B	204\$:	.WORD	205\$-204\$,- 206\$-204\$,- 207\$-204\$,- 208\$-204\$,- 209\$-204\$	
	001E	0014	0032		00873				
		0000'	CF	9F	00875	205\$:	PUSHAB	P.AIF	

		0000'	CF	9F	00879		PUSHAB	P.AIE		
			26	11	0087D		BRB	210\$		
		0000'	CF	9F	0087F	206\$:	PUSHAB	P.AIH		
		0000'	CF	9F	00883		PUSHAB	P.AIG		
			1C	11	00887		BRB	210\$		
		0000'	CF	9F	00889	207\$:	PUSHAB	P.AIJ		
		0000'	CF	9F	0088D		PUSHAB	P.AII		
			12	11	00891		BRB	210\$		
		0000'	CF	9F	00893	208\$:	PUSHAB	P.AIL		
		0000'	CF	9F	00897		PUSHAB	P.AIK		
			08	11	0089B		BRB	210\$		
		0000'	CF	9F	0089D	209\$:	PUSHAB	P.AIN		
		0000'	CF	9F	008A1		PUSHAB	P.AIM		
			53	DD	008A5	210\$:	PUSHL	R3		
F71B	CF		03	FB	008A7		CALLS	#3, APPEND_STRINGS		
	02		54	D1	008AC	211\$:	CMPL	PARAM_TYPE, #2		1329
			1A	12	008AF		BNEQ	212\$		
			57	D4	008B1		CLRL	R7		
		0000'	CF	9F	008B3		PUSHAB	P.AIO		1330
			53	DD	008B7		PUSHL	R3		
F709	CF		02	FB	008B9		CALLS	#2, APPEND_STRINGS		
		0C	AC	DD	008BE		PUSHL	MAXLEN		1331
		04	AE	DD	008C1		PUSHL	4(SP)		
			53	DD	008C4		PUSHL	R3		
0000V	CF		03	FB	008C6		CALLS	#3, UNKNOWN_PARAM_DATA		
	03		54	D1	008CB	212\$:	CMPL	PARAM_TYPE, #3		1332
			10	12	008CE		BNEQ	213\$		
			57	D4	008D0		CLRL	R7		
	7E		58	9A	008D2		MOVZBL	BYTE1, -(SP)		
		0000'	CF	9F	008D5		PUSHAB	P.AIP		
			53	DD	008D9		PUSHL	R3		
F6E7	CF		03	FB	008DB		CALLS	#3, APPEND_STRINGS		
	04		54	D1	008E0	213\$:	CMPL	PARAM_TYPE, #4		1333
			10	12	008E3		BNEQ	214\$		
			57	D4	008E5		CLRL	R7		
	7E		58	9A	008E7		MOVZBL	BYTE1, -(SP)		
		0000'	CF	9F	008EA		PUSHAB	P.AIQ		
			53	DD	008EE		PUSHL	R3		
F6D2	CF		03	FB	008F0		CALLS	#3, APPEND_STRINGS		
	05		54	D1	008F5	214\$:	CMPL	PARAM_TYPE, #5		1334
			46	12	008F8		BNEQ	221\$		
			57	D4	008FA		CLRL	R7		
		0000'	CF	9F	008FC		PUSHAB	P.AIR		
			53	DD	00900		PUSHL	R3		
F6C0	CF		02	FB	00902		CALLS	#2, APPEND_STRINGS		
			58	8F	00907		CASEB	BYTE1, #0, #3		1339
			0008		00908	215\$:	.WORD	216\$-215\$,-		
								217\$-215\$,-		
								218\$-215\$,-		
								219\$-215\$		
		0000'	CF	9F	00913	216\$:	PUSHAB	P.AIT		
		0000'	CF	9F	00917		PUSHAB	P.AIS		
			1C	11	0091B		BRB	220\$		
		0000'	CF	9F	0091D	217\$:	PUSHAB	P.AIV		
		0000'	CF	9F	00921		PUSHAB	P.AIU		
			12	11	00925		BRB	220\$		
		0000'	CF	9F	00927	218\$:	PUSHAB	P.AIX		

0026

03
001C

00
0012

0008

215\$:

1339

		0000'	CF	9F	0092B		PUSHAB	P.AIW		
		08	11	0092F		BRB	220\$			
		0000'	CF	9F	00931	219\$:	PUSHAB	P.AIZ		
		0000'	CF	9F	00935		PUSHAB	P.AIY		
		53	DD	00939	220\$:	PUSHL	R3			
F687	CF	03	FB	0093B		CALLS	#3, APPEND_STRINGS			
	06	54	D1	00940	221\$:	CMPL	PARAM_TYPE, #6			1340
		10	12	00943		BNEQ	222\$			
		57	D4	00945		CLRL	R7			
	7E	58	9A	00947		MOVZBL	BYTE1, -(SP)			
		0000'	CF	9F	0094A		PUSHAB	P.AJA		
		53	DD	0094E		PUSHL	R3			
F672	CF	03	FB	00950		CALLS	#3, APPEND_STRINGS			
	07	54	D1	00955	222\$:	CMPL	PARAM_TYPE, #7			1341
		10	12	00958		BNEQ	223\$			
		57	D4	0095A		CLRL	R7			
	7E	6B	3C	0095C		MOVZWL	(R11), -(SP)			
		0000'	CF	9F	0095F		PUSHAB	P.AJB		
		53	DD	00963		PUSHL	R3			
F65D	CF	03	FB	00965		CALLS	#3, APPEND_STRINGS			
	08	54	D1	0096A	223\$:	CMPL	PARAM_TYPE, #8			1342
		10	12	0096D		BNEQ	224\$			
		57	D4	0096F		CLRL	R7			
	7E	6B	3C	00971		MOVZWL	(R11), -(SP)			
		0000'	CF	9F	00974		PUSHAB	P.AJC		
		53	DD	00978		PUSHL	R3			
F648	CF	03	FB	0097A		CALLS	#3, APPEND_STRINGS			
	09	54	D1	0097F	224\$:	CMPL	PARAM_TYPE, #9			1343
		0F	12	00982		BNEQ	225\$			
		57	D4	00984		CLRL	R7			
		58	DD	00986		PUSHL	R11			
		0000'	CF	9F	00988		PUSHAB	P.AJD		
		53	DD	0098C		PUSHL	R3			
F634	CF	03	FB	0098E		CALLS	#3, APPEND_STRINGS			
	0A	54	D1	00993	225\$:	CMPL	PARAM_TYPE, #10			1344
		2E	12	00996		BNEQ	230\$			
		57	D4	00998		CLRL	R7			
		0000'	CF	9F	0099A		PUSHAB	P.AJE		
		53	DD	0099E		PUSHL	R3			
	01	02	FB	009A0		CALLS	#2, APPEND_STRINGS			
		58	8F	009A5		CASEB	BYTE1, #0, #1			
	000E	0004		009A9	226\$:	.WORD	227\$-226\$, -			1347
							228\$-226\$			
		0000'	CF	9F	009AD	227\$:	PUSHAB	P.AJG		
		0000'	CF	9F	009B1		PUSHAB	P.AJF		
		08	11	009B5		BRB	229\$			
		0000'	CF	9F	009B7	228\$:	PUSHAB	P.AJI		
		0000'	CF	9F	009BB		PUSHAB	P.AJH		
		53	DD	009BF	229\$:	PUSHL	R3			
F601	CF	03	FB	009C1		CALLS	#3, APPEND_STRINGS			
	0B	54	D1	009C6	230\$:	CMPL	PARAM_TYPE, #11			1348
		0D	12	009C9		BNEQ	231\$			
		57	D4	009CB		CLRL	R7			
		0000'	CF	9F	009CD		PUSHAB	P.AJJ		
		53	DD	009D1		PUSHL	R3			
F5EF	CF	02	FB	009D3		CALLS	#2, APPEND_STRINGS			
	0C	54	D1	009D8	231\$:	CMPL	PARAM_TYPE, #12			1349

			0000'	0D	12	009DB		BNEQ	232\$		
				57	D4	009DD		CLRL	R7		
				CF	9F	009DF		PUSHAB	P.AJK		
				53	DD	009E3		PUSHL	R3		
	F5DD	CF		02	FB	009E5		CALLS	#2, APPEND_STRINGS		
		0B		54	D1	009EA	232\$:	CMPL	PARAM_TYPE, #11		1350
				54	1F	009ED		BLSSU	238\$		
		0C		54	D1	009EF		CMPL	PARAM_TYPE, #12		
				2F	1A	009F2		BGTRU	238\$		
				57	D4	009F4		CLRL	R7		
	02	00		58	BF	009F6		CASEB	BYTE1, #0, #2		1351
001A		0010	0006			009FA	233\$:	.WORD	234\$-233\$,-		
									235\$-233\$,-		
									236\$-233\$		
			0000'	CF	9F	00A00	234\$:	PUSHAB	P.AJM		
			0000'	CF	9F	00A04		PUSHAB	P.AJL		
				12	11	00A08		BRB	237\$		
			0000'	CF	9F	00A0A	235\$:	PUSHAB	P.AJO		
			0000'	CF	9F	00A0E		PUSHAB	P.AJN		
				08	11	00A12		BRB	237\$		
			0000'	CF	9F	00A14	236\$:	PUSHAB	P.AJQ		
			0000'	CF	9F	00A18		PUSHAB	P.AJP		
	F5A4	CF		53	DD	00A1C	237\$:	PUSHL	R3		
		0D		03	FB	00A1E		CALLS	#3, APPEND_STRINGS		
				54	D1	00A23	238\$:	CMPL	PARAM_TYPE, #13		1352
				10	12	00A26		BNEQ	239\$		
				57	D4	00A28		CLRL	R7		
		7E		6B	3C	00A2A		MOVZWL	(R11), -(SP)		
			0000'	CF	9F	00A2D		PUSHAB	P.AJR		
				53	DD	00A31		PUSHL	R3		
	F58F	CF		03	FB	00A33		CALLS	#3, APPEND_STRINGS		
		0E		54	D1	00A38	239\$:	CMPL	PARAM_TYPE, #14		1353
				10	12	00A3B		BNEQ	240\$		
				57	D4	00A3D		CLRL	R7		
		7E		58	9A	00A3F		MOVZBL	BYTE1, -(SP)		
			0000'	CF	9F	00A42		PUSHAB	P.AJS		
				53	DD	00A46		PUSHL	R3		
	F57A	CF		03	FB	00A48		CALLS	#3, APPEND_STRINGS		
		0F		54	D1	00A4D	240\$:	CMPL	PARAM_TYPE, #15		1354
				10	12	00A50		BNEQ	241\$		
				57	D4	00A52		CLRL	R7		
		7E		58	9A	00A54		MOVZBL	BYTE1, -(SP)		
			0000'	CF	9F	00A57		PUSHAB	P.AJT		
				53	DD	00A5B		PUSHL	R3		
	F565	CF		03	FB	00A5D		CALLS	#3, APPEND_STRINGS		
		10		54	D1	00A62	241\$:	CMPL	PARAM_TYPE, #16		1355
				03	13	00A65		BEQL	242\$		
			00B2		31	00A67		BRW	258\$		
				57	D4	00A6A	242\$:	CLRL	R7		
			0000'	CF	9F	00A6C		PUSHAB	P.AJU		
				53	DD	00A70		PUSHL	R3		
	F550	CF		02	FB	00A72		CALLS	#2, APPEND_STRINGS		
				58	BF	00A77		CASEB	BYTE1, #0, #12		1369
						00A7B	243\$:	.WORD	244\$-243\$,-		
						00A83			245\$-243\$,-		
						00A8B			246\$-243\$,-		
						00A93			247\$-243\$,-		

0038
0060
0088

0C
002E
0056
007E

0024
004C
0074

						248\$-243\$		
						249\$-243\$		
						250\$-243\$		
						251\$-243\$		
						252\$-243\$		
						253\$-243\$		
						254\$-243\$		
						255\$-243\$		
						256\$-243\$		
		0000'	CF	9F	00A95	244\$:	PUSHAB	P.AJW
		0000'	CF	9F	00A99		PUSHAB	P.AJV
			76	11	00A9D		BRB	257\$
		0000'	CF	9F	00A9F	245\$:	PUSHAB	P.AJY
		0000'	CF	9F	00AA3		PUSHAB	P.AJX
			6C	11	00AA7		BRB	257\$
		0000'	CF	9F	00AA9	246\$:	PUSHAB	P.AKA
		0000'	CF	9F	00AAD		PUSHAB	P.AJZ
			62	11	00AB1		BRB	257\$
		0000'	CF	9F	00AB3	247\$:	PUSHAB	P.AKC
		0000'	CF	9F	00AB7		PUSHAB	P.AKB
			58	11	00ABB		BRB	257\$
		0000'	CF	9F	00ABD	248\$:	PUSHAB	P.AKE
		0000'	CF	9F	00AC1		PUSHAB	P.AKD
			4E	11	00AC5		BRB	257\$
		0000'	CF	9F	00AC7	249\$:	PUSHAB	P.AKG
		0000'	CF	9F	00ACB		PUSHAB	P.AKF
			44	11	00ACF		BRB	257\$
		0000'	CF	9F	00AD1	250\$:	PUSHAB	P.AKI
		0000'	CF	9F	00AD5		PUSHAB	P.AKH
			3A	11	00AD9		BRB	257\$
		0000'	CF	9F	00ADB	251\$:	PUSHAB	P.AKK
		0000'	CF	9F	00ADF		PUSHAB	P.AKJ
			30	11	00AE3		BRB	257\$
		0000'	CF	9F	00AE5	252\$:	PUSHAB	P.AKM
		0000'	CF	9F	00AE9		PUSHAB	P.AKL
			26	11	00AED		BRB	257\$
		0000'	CF	9F	00AEF	253\$:	PUSHAB	P.AKO
		0000'	CF	9F	00AF3		PUSHAB	P.AKN
			1C	11	00AF7		BRB	257\$
		0000'	CF	9F	00AF9	254\$:	PUSHAB	P.AKQ
		0000'	CF	9F	00AFD		PUSHAB	P.AKP
			12	11	00B01		BRB	257\$
		0000'	CF	9F	00B03	255\$:	PUSHAB	P.AKS
		0000'	CF	9F	00B07		PUSHAB	P.AKR
			08	11	00B0B		BRB	257\$
		0000'	CF	9F	00B0D	256\$:	PUSHAB	P.AKU
		0000'	CF	9F	00B11		PUSHAB	P.AKT
			53	DD	00B15	257\$:	PUSHL	R3
F4AB	CF		03	FB	00B17		CALLS	#3, APPEND_STRINGS
	11		54	D1	00B1C	258\$:	CMPL	PARAM_TYPE, #17
			10	12	00B1F		BNEQ	259\$
			57	D4	00B21		CLRL	R7
	7E		68	3C	00B23		MOVZWL	(R11), -(SP)
		0000'	CF	9F	00B26		PUSHAB	P.AKV
			53	DD	00B2A		PUSHL	R3
F496	CF		03	FB	00B2C		CALLS	#3, APPEND_STRINGS
	12		54	D1	00B31	259\$:	CMPL	PARAM_TYPE, #18

1370

1371

			1A 12 00B34	BNEQ	260\$	
			57 D4 00B36	CLRL	R7	
		0000'	CF 9F 00B38	PUSHAB	P.AKW	
F484	CF		53 DD 00B3C	PUSHL	R3	
		0C	02 FB 00B3E	CALLS	#2, APPEND_STRINGS	
		04	AC DD 00B43	PUSHL	MAXLEN	1372
			AE DD 00B46	PUSHL	4(SP)	
0000V	CF		53 DD 00B49	PUSHL	R3	
	13		03 FB 00B4B	CALLS	#3, UNKNOWN_PARAM_DATA	
			54 D1 00B50	CMPL	PARAM_TYPE, #19	1373
			1A 12 00B53	BNEQ	261\$	
		0000'	57 D4 00B55	CLRL	R7	
			CF 9F 00B57	PUSHAB	P.AKX	
F465	CF		53 DD 00B5B	PUSHL	R3	
		0C	02 FB 00B5D	CALLS	#2, APPEND_STRINGS	
		04	AC DD 00B62	PUSHL	MAXLEN	1374
			AE DD 00B65	PUSHL	4(SP)	
0000V	CF		53 DD 00B68	PUSHL	R3	
			03 FB 00B6A	CALLS	#3, UNKNOWN_PARAM_DATA	
	06	00F0	31 00B6F	BRW	277\$	1376
			55 D1 00B72	CMPL	R5, #6	1379
			46 12 00B75	BNEQ	268\$	
	57		01 D0 00B77	MOVL	#1, R7	1380
			54 D5 00B7A	TSTL	PARAM_TYPE	1383
			10 12 00B7C	BNEQ	263\$	
			57 D4 00B7E	CLRL	R7	
	7E		6B 3C 00B80	MOVZWL	(R11), -(SP)	
		0000'	CF 9F 00B83	PUSHAB	P.AKY	
			53 DD 00B87	PUSHL	R3	
F439	CF		03 FB 00B89	CALLS	#3, APPEND_STRINGS	
	01		54 D1 00B8E	CMPL	PARAM_TYPE, #1	1384
			DC 12 00B91	BNEQ	261\$	
		0000'	57 D4 00B93	CLRL	R7	
			CF 9F 00B95	PUSHAB	P.AKZ	
			53 DD 00B99	PUSHL	R3	
01	F427	CF	02 FB 00B9B	CALLS	#2, APPEND_STRINGS	
	00		58 8F 00BA0	CASEB	BYTE1, #0, #1	1385
	000E	0004	00BA4	.WORD	265\$-264\$,-	
					266\$-264\$	
		0000'	CF 9F 00BAB	PUSHAB	P.ALB	
		0000'	CF 9F 00BAC	PUSHAB	P.ALA	
			08 11 00BB0	BRB	267\$	
		0000'	CF 9F 00BB2	PUSHAB	P.ALD	
		0000'	CF 9F 00BB6	PUSHAB	P.ALC	
			009E 31 00BBA	BRW	276\$	
00000045	8F		55 D1 00BBD	CMPL	R5, #69	1389
			12 13 00BC4	BEQL	269\$	
000000E1	8F		55 D1 00BC6	CMPL	R5, #225	
			19 1F 00BCD	BLSSU	270\$	
000000E2	8F		55 D1 00BCF	CMPL	R5, #226	
			10 1A 00BD6	BGTRU	270\$	
		0C	AC DD 00BD9	PUSHL	MAXLEN	1393
		04	AE DD 00BDB	PUSHL	4(SP)	
			53 DD 00BDE	PUSHL	R3	
0000V	CF		03 FB 00BE0	CALLS	#3, UNKNOWN_PARAM_DATA	
		0089	31 00BE5	BRW	279\$	
00000080	8F		55 D1 00BE8	CMPL	R5, #128	1395

		74	12	00BEF	BNEQ	278\$	
	57	01	DD	00BF1	MOVL	#1, R7	1396
		54	D5	0CBF4	TSTL	PARAM_TYPE	1399
		2E	12	00BF6	BNEQ	273\$	
		57	D4	00BF8	CLRL	R7	
		CF	9F	00BFA	PUSHAB	P, ALE	
		53	DD	00BFE	PUSHL	R3	
F3C2	CF	02	FB	00C00	CALLS	#2, APPEND_STRINGS	
		6B	95	00C05	TSTB	(R11)	1400
		10	12	00C07	BNEQ	271\$	
	7E	69	3C	00C09	MOVZWL	(R9), -(SP)	1402
0000V	CF	01	FB	00C0C	CALLS	#1, FORMAT_NODEADR	
		50	DD	00C11	PUSHL	R0	
		CF	9F	00C13	PUSHAB	P, ALF	
		06	11	00C17	BRB	272\$	
		5B	DD	00C19	PUSHL	R11	1404
		CF	9F	00C1B	PUSHAB	P, ALG	
F3A1	CF	53	DD	00C1F	PUSHL	R3	
	01	03	FB	00C21	CALLS	#3, APPEND_STRINGS	
		54	D1	00C26	CMPL	PARAM_TYPE, #1	1405
		0F	12	00C29	BNEQ	274\$	
		57	D4	00C2B	CLRL	R7	
		5B	DD	00C2D	PUSHL	R11	
		CF	9F	00C2F	PUSHAB	P, ALH	
F38D	CF	53	DD	00C33	PUSHL	R3	
	02	03	FB	00C35	CALLS	#3, APPEND_STRINGS	
		54	D1	00C3A	CMPL	PARAM_TYPE, #2	1406
		0F	12	00C3D	BNEQ	275\$	
		57	D4	00C3F	CLRL	R7	
		6B	DD	00C41	PUSHL	(R11)	
		CF	9F	00C43	PUSHAB	P, ALI	
F379	CF	53	DD	00C47	PUSHL	R3	
	03	03	FB	00C49	CALLS	#3, APPEND_STRINGS	
		54	D1	00C4E	CMPL	PARAM_TYPE, #3	1407
		0F	12	00C51	BNEQ	277\$	
		57	D4	00C53	CLRL	R7	
		6B	DD	00C55	PUSHL	(R11)	
		CF	9F	00C57	PUSHAB	P, ALJ	
F365	CF	53	DD	00C5B	PUSHL	R3	
	0C	03	FB	00C5D	CALLS	#3, APPEND_STRINGS	
		57	E9	00C62	BLBC	R7, 279\$	1408
		0C	AC	DD	00C65	PUSHL	MAXLEN
		52	DD	00C68	PUSHL	PTR	1411
		53	DD	00C6A	PUSHL	R3	
0000V	CF	03	FB	00C6C	CALLS	#3, UNKNOWN_PARAMETER	
	08	56	D0	00C71	MOVL	NEXTPTR, @PARAM	1414
		04	00C75	RET			1416

; Routine Size: 3190 bytes, Routine Base: \$CODE\$ + 029C

```

1014 1417 1 ROUTINE unknown_parameter (string, ptr, maxlen): NOVALUE =
1015 1418 1
1016 1419 1 ---
1017 1420 1
1018 1421 1     Append a unknown parameter to a given string.
1019 1422 1
1020 1423 1 Inputs:
1021 1424 1
1022 1425 1     desc = Address of current string buffer
1023 1426 1     ptr = Address of parameter (data id, data type, ...)
1024 1427 1     maxlen = Maximum length that the parameter may be
1025 1428 1
1026 1429 1 Outputs:
1027 1430 1
1028 1431 1     The string is updated.
1029 1432 1 ---
1030 1433 1
1031 1434 1 BEGIN
1032 1435 1
1033 1436 1 MAP
1034 1437 1     ptr:          REF $BBLOCK;          ! Address of parameter data
1035 1438 1
1036 1439 1 BIND
1037 1440 1     desc = .string: VECTOR;          ! Address of string descriptor
1038 1441 1
1039 1442 1 append('Parameter #!UL = ', .ptr [0,0,11,0]); ! Show parameter type
1040 1443 1
1041 1444 1 unknown_param_data(desc, .ptr+2, .maxlen);    ! Append unknown data
1042 1445 1
1043 1446 1 END;

```

```

.PSECT $SPLITS,NOWRT,NOEXE,2
4C 55 21 23 20 72 65 74 65 6D 61 72 61 50 11 01146 P.ALK: .ASCII <17>\Parameter #!UL = \
20 3D 20 01155

```

```

.PSECT $CODE$,NOWRT,2
0000 0000 UNKNOWN_PARAMETER:
7E 08 BC 0B 00 EF 00002 .WORD Save nothing : 1417
0000' CF 9F 00008 EXTZV #0, #11, @PTR, -(SP) : 1442
04 AC DD 0000C PUSHAB P.ALK
F33D CF 03 FB 0000F PUSHL STRING
0C AC DD 00014 CALLS #3, APPEND_STRINGS
7E 08 AC 02 C1 00017 PUSHL MAXLEN : 1444
04 AC DD 0001C ADDL3 #2, PTR, -(SP)
0000V CF 03 FB 0001F PUSHL STRING
04 00024 CALLS #3, UNKNOWN_PARAM_DATA : 1446
RET

```

; Routine Size: 37 bytes. Routine Base: \$CODE\$ + 0F12

```

1045 1447 1 ROUTINE unknown_param_data (string, param, maxlen): NOVALUE =
1046 1448 1
1047 1449 1 ---
1048 1450 1
1049 1451 1     Append a unknown parameter to a given string.
1050 1452 1
1051 1453 1 Inputs:
1052 1454 1
1053 1455 1     desc = Address of current string buffer
1054 1456 1     param = Address of parameter data type
1055 1457 1     maxlen = Maximum length that the parameter may be
1056 1458 1
1057 1459 1 Outputs:
1058 1460 1
1059 1461 1     The string is updated.
1060 1462 1 ---
1061 1463 1
1062 1464 2 BEGIN
1063 1465 2
1064 1466 2 BIND
1065 1467 2     desc = .string: VECTOR;           ! Address of string descriptor
1066 1468 2
1067 1469 2 LOCAL
1068 1470 2     ptr:          REF $BBLOCK,         ! Address of parameter data
1069 1471 2     fields;        ! Number of fields in parameter
1070 1472 2
1071 1473 2     ptr = .param;          ! Point to first data type
1072 1474 2
1073 1475 2     IF .ptr [0,7,1,0]      ! If coded
1074 1476 2         AND .ptr [0,6,1,0] ! and multiple number of fields,
1075 1477 2     THEN
1076 1478 2         BEGIN
1077 1479 2             fields = .ptr [0,0,6,0]; ! Get # fields in parameter
1078 1480 2             ptr = .ptr + 1;          ! and skip to actual first data item
1079 1481 2         END
1080 1482 2     ELSE
1081 1483 2         fields = 1;           ! Else, indicate only one field
1082 1484 2
1083 1485 2     INCR i FROM 1 TO .fields ! For each field
1084 1486 2     DO
1085 1487 2         BEGIN
1086 1488 2             IF .ptr [0,7,1,0] ! If coded
1087 1489 2             THEN
1088 1490 2                 BEGIN
1089 1491 2                     INCRU j FROM 1 TO .ptr [0,0,6,0] ! For each byte in coded field
1090 1492 2                     DO
1091 1493 2                         append('!XB ' .ptr [.j,0,8,0]); ! append the hex for that byte
1092 1494 2                         desc [0] = .desc [0] - 1; ! Strip off the last blank
1093 1495 2                         ptr = .ptr + 1 + .ptr [0,0,6,0];
1094 1496 2                     END
1095 1497 2             ELSE
1096 1498 2                 IF .ptr [0,6,1,0] ! If ASCII image field
1097 1499 2                 THEN
1098 1500 2                     BEGIN
1099 1501 2                         append('!AC' .ptr+1); ! append the string
1100 1502 2                         ptr = .ptr + 2 + CH$RCHAR(.ptr+1); ! Use ASCII length
1101 1503 2                     END

```

```

1102 1504 3 ELSE
1103 1505 4 BEGIN
1104 1506 5 LOCAL bytes, longword;
1105 1507 6 bytes = .ptr [0,0,4,0]; ! # bytes in field
1106 1508 7 IF .bytes EQL 0 ! If zero,
1107 1509 8 THEN
1108 1510 9 BEGIN ! then treat as hex image (counted)
1109 1511 10 INCRU j FROM 1 TO CH$RCHAR(.ptr+1) ! For each byte in field
1110 1512 11 DO
1111 1513 12 append('!XB', .ptr [j+1,0,8,0]); ! append the hex for that byte
1112 1514 13 ptr = .ptr + 2 + CH$RCHAR(.ptr+1);
1113 1515 14 END
1114 1516 15 ELSE BEGIN
1115 1517 16 IF .bytes GTRU 4 ! If greater than longword,
1116 1518 17 THEN
1117 1519 18 bytes = 4; ! then truncate to longword
1118 1520 19 longword = .ptr [1,0,8*.bytes,0]; ! Extract unsigned to longword
1119 1521 20 SELECTONEU .ptr [0,4,2,0] ! Based on output format,
1120 1522 21 OF
1121 1523 22 SET
1122 1524 23 [0]: append('!UL', .longword); ! Unsigned decimal
1123 1525 24 [1]: append('!SL', .ptr [1,0,8*.bytes,1]); ! Signed decimal
1124 1526 25 [2]: SELECTONEU .bytes OF SET ! Hexidecimal
1125 1527 26 [1]: append('!XB', .longword); ! byte
1126 1528 27 [2]: append('!XW', .longword); ! word
1127 1529 28 [4, OTHERWISE]: append('!XL', .longword); ! longword
1128 1530 29 TES;
1129 1531 30 [3]: SELECTONEU .bytes OF SET ! Octal
1130 1532 31 [1]: append('!OB', .longword); ! byte
1131 1533 32 [2]: append('!OW', .longword); ! word
1132 1534 33 [4, OTHERWISE]: append('!OL', .longword); ! longword
1133 1535 34 TES;
1134 1536 35 TES;
1135 1537 36 ptr = .ptr + 1 + .ptr [0,0,4,0];
1136 1538 37 END;
1137 1539 38 END;
1138 1540 39
1139 1541 40 IF .ptr - .param GTRU .maxlen ! If exceeded maximum allowable length
1140 1542 41 THEN
1141 1543 42 BEGIN
1142 1544 43 SIGNAL(msg(badparam)); ! Signal illegal NICE parameter
1143 1545 44 RETURN;
1144 1546 45 END;
1145 1547 46
1146 1548 47 IF .i LSS .fields ! If not last field,
1147 1549 48 THEN
1148 1550 49 append(' '); ! then delimit fields with a blank
1149 1551 50 END;
1150 1552 51
1151 1553 52 END;

```

.PSECT \$SPLITS, NOWRT, NOEXE, 2

20 42 58 21 04 01158 P.ALL: .ASCII <4>\!XB \

43 41 21 03 01150 P.ALM: .ASCII <3>\!AC \

:

42	58	21	03	01161	P.ALN:	.ASCII	<3>\XB\	
4C	55	21	03	01165	P.ALC:	.ASCII	<3>\UL\	
4C	53	21	03	01169	P.ALP:	.ASCII	<3>\SL\	
42	58	21	03	0116D	P.ALQ:	.ASCII	<3>\XB\	
57	58	21	03	01171	P.ALK:	.ASCII	<3>\XW\	
4C	58	21	03	01175	P.ALS:	.ASCII	<3>\XL\	
42	4F	21	03	01179	P.ALT:	.ASCII	<3>\OB\	
57	4F	21	03	0117D	P.ALU:	.ASCII	<3>\OW\	
4C	4F	21	03	01181	P.ALV:	.ASCII	<3>\OL\	
		20	01	01185	P.ALW:	.ASCII	<1>\ \	

.EXTRN EVLS_BADPARAM

.PSECT \$CODE\$,NOWRT,2

OFFC 00000 UNKNOWN_PARAM_DATA:

		58	0000'	CF	9E	00002	.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	1447
		58	04	AC	D0	00007	MOVAB	P.ALL, R11	
		52	08	AC	D0	0000B	MOVL	STRING, R8	1467
				62	95	0000F	MOVL	PARAM, PTR	1473
				0B	18	00011	TSTB	(PTR)	1475
5A	07	62		06	E1	00013	BGEQ	1\$	
	82	06		00	EF	00017	BBC	#6, (PTR), 1\$	1476
				03	11	0001C	EXTZV	#0, #6, (PTR)+, FIELDS	1479
		5A		01	D0	0001E	BRB	2\$	1475
				59	D4	00021	MOVL	#1, FIELDS	1483
			0125	31	00023		CLRL	I	1541
			62	95	00026	3\$:	BRW	24\$	
			27	18	00028		TSTB	(PTR)	1488
53	62	06		00	EF	0002A	BGEQ	6\$	
		54		01	D0	0002F	EXTZV	#0, #6, (PTR), R3	1491
				0F	11	00032	MOVL	#1, J	
		7E		6442	9A	00034	BRB	5\$	
			0900	8F	BB	00038	MOVZBL	(J)[PTR], -(SP)	1493
		F2EB	CF	03	FB	0003C	PUSHR	#*M<R8,R11>	
				54	D6	00041	CALLS	#3, APPEND_STRINGS	
		53		54	D1	00043	INCL	J	
				EC	1B	00046	CMPL	J, R3	
				68	D7	00048	BLEQU	4\$	
		52	01	A342	9E	0004A	DECL	(R8)	1494
				4B	11	0004F	MOVAB	1(R3)[PTR], PTR	1495
		57	01	A2	9E	00051	BRB	10\$	1488
	16	62		06	E1	00055	MOVAB	1(PTR), R7	1501
				57	DD	00059	BBC	#6, (PTR), 7\$	1498
				05	AB	0005B	PUSHL	R7	1501
				58	DD	0005E	PUSHAB	P.ALW	
		F2C7	CF	03	FB	00060	PUSHL	R8	
		50		67	9A	00065	CALLS	#3, APPEND_STRINGS	
		52	02	A042	9E	00068	MOVZBL	(R7), R0	1502
				2D	11	0006D	MOVAB	2(R0)[PTR], PTR	
56	62	04		00	EF	0006F	BRB	10\$	1498
		53		56	D0	00074	EXTZV	#0, #4, (PTR), R6	1507
				26	12	00077	MOVL	R6, BYTES	
		55		67	9A	00079	BNEQ	11\$	1508
		54		01	D0	0007C	MOVZBL	(R7), R5	1511
				11	11	0007F	MOVL	#1, J	
							BRB	9\$	

		7E	01	A442	9A	00081	8\$:	MOVZBL	1(J)[PTR], -(SP)	1513
			09	AB	9F	00086		PUSHAB	P.ALN	
				58	DD	00089		PUSHL	R8	
	F29C	CF		03	FB	0008B		CALLS	#3, APPEND_STRINGS	
				54	D6	00090		INCL	J	
		55		54	D1	00092	9\$:	CMPL	J, R5	
				EA	1B	00095		BLEQU	8\$	
		52	02	A542	9E	00097		MOVAB	2(R5)[PTR], PTR	1514
				0084	31	0009C	10\$:	BRW	22\$	1508
		04		53	D1	0009F	11\$:	CMPL	BYTES, #4	1517
				03	1B	000A2		BLEQU	12\$	
		53		04	D0	000A4		MOVL	#4, BYTES	1519
	54			03	7B	000A7	12\$:	ASHL	#3, BYTES, R4	1520
55	67			00	EF	000AB		EXTZV	#0, R4, (R7), LONGWORD	
				62	93	000B0		BITB	(PTR), #48	1524
		30		07	12	000B3		BNEQ	13\$	
				55	DD	000B5		PUSHL	LONGWORD	
			0D	AB	9F	000B7		PUSHAB	P.ALO	
				5B	11	000BA		BRB	20\$	
01	62	02		04	ED	000BC	13\$:	CMPZV	#4, #2, (PTR), #1	1525
				0A	12	000C1		BNEQ	14\$	
7E	67	54		00	EE	000C3		EXTV	#0, R4, (R7), -(SP)	
			11	AB	9F	000C8		PUSHAB	P.ALP	
				4A	11	000CB		BRB	20\$	
02	62	02		04	ED	000CD	14\$:	CMPZV	#4, #2, (PTR), #2	1526
				1F	12	000D2		BNEQ	17\$	
		01		53	D1	000D4		CMPL	BYTES, #1	1527
				07	12	000D7		BNEQ	15\$	
				55	DD	000D9		PUSHL	LONGWORD	
			15	AB	9F	000DB		PUSHAB	P.ALQ	
				37	11	000DE		BRB	20\$	
		02		53	D1	000E0	15\$:	CMPL	BYTES, #2	1528
				07	12	000E3		BNEQ	16\$	
				55	DD	000E5		PUSHL	LONGWORD	
			19	AB	9F	000E7		PUSHAB	P.ALR	
				2B	11	000EA		BRB	20\$	
				55	DD	000EC	16\$:	PUSHL	LONGWORD	1529
			1D	AB	9F	000EE		PUSHAB	P.ALS	
				24	11	000F1		BRB	20\$	
03	62	02		04	ED	000F3	17\$:	CMPZV	#4, #2, (PTR), #3	1531
				24	12	000F8		BNEQ	21\$	
		01		53	D1	000FA		CMPL	BYTES, #1	1532
				07	12	000FD		BNEQ	18\$	
				55	DD	000FF		PUSHL	LONGWORD	
			21	AB	9F	00101		PUSHAB	P.ALT	
				11	11	00104		BRB	20\$	
		02		53	D1	00106	18\$:	CMPL	BYTES, #2	1533
				07	12	00109		BNEQ	19\$	
				55	DD	0010B		PUSHL	LONGWORD	
			25	AB	9F	0010D		PUSHAB	P.ALU	
				05	11	00110		BRB	20\$	
				55	DD	00112	19\$:	PUSHL	LONGWORD	1534
			29	AB	9F	00114		PUSHAB	P.ALV	
				58	DD	00117	20\$:	PUSHL	R8	
				03	FB	00119		CALLS	#3, APPEND_STRINGS	
	F20E	CF	01	A642	9E	0011E	21\$:	MOVAB	1(R6)[PTR], PTR	1537
		52		08	AC	C3	22\$:	SUBL3	PARAM, PTR, R0	1541
	50	52								

CONSOLE
V04-000

D 2
16-Sep-1984 01:31:10
14-Sep-1984 12:28:46

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EVL.SRC]CONSOLE.B32;1

Page 63
(7)

OC	AC	50	D1	00128	CMPL	R0, MAXLEN	:
		0E	1B	0012C	BLEQU	23\$:
00000000G	00	8F	DD	0012E	PUSHL	#EVL\$ BADPARAM	1544
		01	FB	00134	CALLS	#1, LTB\$SIGNAL	:
	5A		04	00138	RET		1543
		59	D1	0013C	CMPL	1, FIELDS	1548
		0A	1B	0013F	BGEQ	24\$:
		AB	9F	00141	PUSHAB	P, ALW	1550
		58	DD	00144	PUSHL	R0	:
FED5	59	F1E1	CF	02	CALLS	#2, APPEND_STRINGS	:
			01	5A	ACBL	FIELDS, #1, 1, 3\$	1485
				04	RET		1553
							:

; Routine Size: 338 bytes, Routine Base: \$CODE\$ + 0F37

```

1153 1554 1 ROUTINE format_counter (string, param, maxlen): NOVALUE =
1154 1555 1
1155 1556 1 ---
1156 1557 1
1157 1558 1     Format a NICE counter and append the
1158 1559 1     descriptive text to a given string.
1159 1560 1
1160 1561 1     Inputs:
1161 1562 1
1162 1563 1     string = Address of descriptor of existing string
1163 1564 1     param  = Address of longword containing pointer to counter
1164 1565 1     maxlen = Maximum length that the parameter may be
1165 1566 1
1166 1567 1     Outputs:
1167 1568 1
1168 1569 1     param = Address of longword pointing to location following counter
1169 1570 1 ---
1170 1571 1
1171 1572 1 BEGIN
1172 1573 1
1173 1574 1 BIND
1174 1575 1     desc = .string: VECTOR;      ! Address of string descriptor
1175 1576 1
1176 1577 1 LOCAL
1177 1578 1     ptr:      REF $BBLOCK,      ! Address of counter
1178 1579 1     type,      ! Counter type
1179 1580 1     count,     ! Actual counter value
1180 1581 1     width;     ! Width of counter field
1181 1582 1
1182 1583 1     ptr = ..param;              ! Get address of counter
1183 1584 1
1184 1585 1     width = 4 * .ptr [0,13,2,0]; ! Get the counter width
1185 1586 1
1186 1587 1     IF .ptr [0,12,1,0]          ! If bit mapped,
1187 1588 1     THEN
1188 1589 1         BEGIN
1189 1590 1             count = .ptr [4,0,..width,0];      ! Get unsigned counter
1190 1591 1             .param = .ptr + 4 + .width/8;      ! Update next position
1191 1592 1         END
1192 1593 1     ELSE
1193 1594 1         BEGIN
1194 1595 1             count = .ptr [2,0,..width,0];      ! Get unsigned counter
1195 1596 1             .param = .ptr + 2 + .width/8;      ! Update next position
1196 1597 1         END;
1197 1598 1
1198 1599 1     append('!10UL ',..count);      ! Append the count value
1199 1600 1
1200 1601 1     type = .ptr [0,0,11,0];        ! Get counter type
1201 1602 1
1202 1603 1     INCRA ptr FROM known_counters BY 6      ! Scan known event table
1203 1604 1     DO
1204 1605 1         BEGIN
1205 1606 1             MAP ptr: REF $BBLOCK;
1206 1607 1             IF ..ptr EQL 0          ! If end of table,
1207 1608 1             THEN
1208 1609 1                 BEGIN
1209 1610 1                     append('Unknown counter type !UL',..type);

```



```

1210      1611      4      EXITLOOP;      ! then exit with failure
1211      1612      4      END;
1212      1613      4      IF .type EQLU .ptr [0,0,16,0]      ! If known counter,
1213      1614      4      THEN
1214      1615      4      BEGIN
1215      1616      4      LOCAL
1216      1617      4      textbuf: VECTOR [128,BYTE], ! Message text buffer
1217      1618      4      text: VECTOR [2]; ! Message text descriptor
1218      1619      4      text [0] = 128; ! Setup descriptor
1219      1620      4      text [1] = textbuf;
1220      1621      4      $GETMSG(MSGID = .ptr [2,0,32,0], ! Get message text
1221      1622      4      BUFADR = text,
1222      1623      4      MSGLEN = text,
1223      1624      4      FLAGS = 1); ! without any prefixing
1224      1625      4      append('!AS', text);
1225      1626      4      EXITLOOP;
1226      1627      4      END;
1227      1628      4      END;
1228      1629      4
1229      1630      1      END;

```

```

.PSECT $SPLITS,NOWRT,NOEXE,2
65 74 6E 75 6F 63 20 20 20 4C 55 30 31 21 07 01187 P.ALX: .ASCII <7>\!10UL \
6E 77 6F 6E 68 6E 55 18 0118F P.ALY: .ASCII <24>\Unknown counter type !UL\
4C 55 21 20 65 70 79 74 20 72 0119E
53 41 21 03 011AB P.ALZ: .ASCII <3>\!AS\

```

```

.PSECT $CODES,NOWRT,2
000C 00000 FORMAT_COUNTER:
5E FF78 CE 9E 00002 .WORD Save R2,R3      1554
52 08 BC D0 00007 MOVAB -136(SP), SP
02 0D EF 0000B MOVL @PARAM, PTR      1583
50 50 50 78 00010 EXTZV #13, #2, (PTR), WIDTH      1585
51 50 08 C7 00014 ASHL WIDTH, #4, WIDTH
51 52 C0 00018 DIVL3 #8, WIDTH, R1      1591
53 04 0D 0C E1 0001B ADDL2 PTR, R1
A2 A2 50 00 EF 0001F BBC #12, (PTR), 1$      1587
08 BC 04 A1 9E 00025 EXTZV #0, WIDTH, 4(PTR), COUNT      1590
08 BC 04 0B 11 0002A MOVAB 4(R1), @PARAM      1591
53 02 A2 50 00 EF 0002C BRB 2$      1587
08 BC 02 A1 9E 00032 EXTZV #0, WIDTH, 2(PTR), COUNT      1595
0000' 04 53 DD 00037 1$: MOVAB 2(R1), @PARAM      1596
0000' 04 CF 9F 00039 2$: PUSHL COUNT      1599
F195 CF 03 FB 00040 PUSHL P.ALX
53 52 00 00 EF 00045 PUSHL STRING
08 53 DD 0004F CALLS #3, APPEND STRINGS      1601
08 12 00051 EXTZV #0, #11, (PTR), TYPE      1603
53 DD 00053 MOVAB KNOWN_COUNTERS, R2      1607
TSTL (PTR)
BNEQ 4$
PUSHL TYPE      1610

```

CONSOLE
V04-000

6 2
16-Sep-1984 01:31:10
14-Sep-1984 12:28:46

VAX-11 BLISS-32 V4.0-742
DISK\$VMSMASTER:[EVL.SRC]CONSOLE.B32;1
Page 66
(8)

53	62	10	0000'	CF	9F	00055	PUSHAB	P.ALZ	:	
				29	11	00059	BRB	5\$:	
				00	ED	0005B	4\$: CMPZV	#0, #16, (PTR), TYPE	:	1613
				2B	12	00060	BNEQ	6\$:	
		6E	80	8F	9A	00062	MOVZBL	#128, TEXT	:	1619
	04	AE	08	AE	9E	00066	MOVAB	TEXTBUF, TEXT+4	:	1620
		7E		01	7D	0006B	MOVQ	#1, -(SP)	:	1624
			08	AE	9F	0006E	PUSHAB	TEXT	:	
			0C	AE	9F	00071	PUSHAB	TEXT	:	
			02	A2	DD	00074	PUSHL	2(PTR)	:	
	00000000G	00		05	FB	00077	CALLS	#5, SYS\$GETMSG	:	
				5E	DD	0007E	PUSHL	SP	:	1625
			0000'	CF	9F	00080	PUSHAB	P.ALZ	:	
			04	AC	DD	00084	5\$: PUSHL	STRING	:	
	F14E	CF		03	FB	00087	CALLS	#3, APPEND_STRINGS	:	
					04	0008C	RET		:	1615
		52		06	CD	0008D	6\$: ADDL2	#6, PTR	:	1603
				BD	11	00090	BRB	3\$:	

; Routine Size: 146 bytes, Routine Base: \$CODE\$ + 1089

		.PSECT	\$OWNS,NOEXE,2	
00000	STRING:	.BLKB	40	
00028	DESC:	.BLKB	8	
		.EXTRN	SYSS\$FAO	
		.PSECT	\$CODE\$,NOWRT,2	
00000		.ENTRY	FORMAT NODEADR, Save R2,R3	1631
00002		MOVAB	SYSS\$FAO, R3	:
00009		MOVAB	DESC, R2	:
0000E		MOVL	#40, DESC	1659
00011		MOVAB	STRING, DESC+4	1660
00016		BITB	ADDRESS+1, #252	1662
0001B		BNEQ	1\$:
0001D		EXTZV	#0, #10, ADDRESS, -(SP)	1666
00023		PUSHL	R2	:
00025		PUSHL	R2	:
00027		PUSHAB	P.AMA	:
0002B		CALLS	#4, SYSS\$FAO	:
0002E		BRB	2\$:
00030	1\$:	EXTZV	#0, #10, ADDRESS, -(SP)	1671
00036		EXTZV	#2, #6, ADDRESS+1, -(SP)	:
0003C		PUSHL	R2	:
0003E		PUSHL	R2	:
00040		PUSHAB	P.AMC	:
00044		CALLS	#5, SYSS\$FAO	:
00047	2\$:	MOVAB	DESC, R0	1673
0004A		RET		1675

; Routine Size: 75 bytes, Routine Base: \$CODES + 111B

: 1277 1676 1 END
: 1278 1677 0 ELUDOM

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

Name	Bytes	Attributes					
\$PLITS	4552	NOVEC,NOWRT,	RD	NOEXE,NOSHR,	LCL,	REL,	CON,NOPIC,ALIGN(2)
\$CODE\$	4454	NOVEC,NOWRT,	RD	EXE,NOSHR,	LCL,	REL,	CON,NOPIC,ALIGN(2)
\$OWNS	48	NOVEC, WRT,	RD	NOEXE,NOSHR,	LCL,	REL,	CON,NOPIC,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	10	0	581	00:01.1
\$255\$DUA28:[SHRLIB]NMALIBRY.L32;1	887	122	13	47	00:00.7
\$255\$DUA28:[EVL.OBJ]EVCDEF.L32;1	213	133	62	15	00:00.1

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:CONSOLE/OBJ=OBJ\$:CONSOLE MSRC\$:CONSOLE/UPDATE=(ENH\$:CONSOLE)

: Size: 4454 code + 4600 data bytes
: Run Time: 01:35.1
: Elapsed Time: 03:25.5
: Lines/CPU Min: 1057
: Lexemes/CPU-Min: 23866
: Memory Used: 1000 pages
: Compilation Complete

EVJULIAN
LIS

EVDEF
LIS

ERRMSG
LIS

EVLBRARY
LIS

EVLTRANS
LIS

EVLSHOW
LIS

EVLMAIN
LIS

RECEIVER
LIS